NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

Finite Memory Model for Haptic Recognition

by

Philip G. Beierl

December 1991

Thesis Advisor Morris R. Driels

Approved for public release; distribution is unlimited.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If Applicable)	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (city, state, and ZIP code) Monterey, CA 93943-5000	7b. ADDRESS (city, state, and ZIP code) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If Applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (city, state, and ZIP code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
FINITE MEMORY MODEL FOR HAPTIC RECOGNITION

12. PERSONAL AUTHOR(S)
Philip G. Beierl

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month,day)	15. PAGE COUNT
Master's Thesis	FROM TO	December 1991	82

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUBGROUP	Haptic Perception Telepresence Memory

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This study attempts to model the process by which humans identify remote objects using a force reflecting telemanipulator in order to apply this understanding to future ROV designs employing the concept of telepresence. A theoretical model is proposed in which object identification is dependent primarily upon feature identification and capacity to remember the sequence of features. A computer simulation of this model is constructed and used to produce theoretical object identification performance which can be compared to actual human performance. The capacity for short term memory of a sequence of features is also studied in a laboratory using a telemanipulator.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT X UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Morris R. Driels	22b. TELEPHONE (Include Area Code) (408) 646-3383 22C. Office Symbol ME/Dr

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted SECURITY CLASSIFICATION OF THIS PAGE
All other editions are obsolete Unclassified

Approved for public release; distribution is unlimited.

Finite Memory Model for Haptic Recognition

by

Philip G. Beierl
Lieutenant Commander, United States Navy
B.S., Massachusetts Institute of Technology, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

December 1991

Author: Philip G. Beierl

Approved by: Morris R. Driels, Thesis Advisor

Anthony J. Healey, Chairman
Department of Mechanical Engineering

ABSTRACT

This study attempts to model the process by which humans identify remote objects using a force-reflecting telemanipulator in order to apply this understanding to future ROV designs employing the concept of telepresence. A theoretical model is proposed in which object identification is dependent primarily upon feature identification and capacity to remember the sequence of features. A computer simulation of this model is constructed and used to produce theoretical object identification performance which can be compared to actual human performance. The capacity for short term memory of a sequence of features is also studied in a laboratory using a telemanipulator.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. TELEPRESENCE AND THE NEXT GENERATION OF REMOTELY OPERATED VEHICLES (ROVS)

1. Current designs and sensory limitations

Remotely operated vehicle (ROV) is the term used to describe a class of mobile machines which operate in a hazardous environment with control exercised from a safe remote location. The most common such environment is found in the sea with the ROV designed to perform various underwater tasks at some distance from a controlling surface vessel, often under conditions in which human divers could not work. ROVs find their main applications in the two areas of search and recovery of lost objects, and inspection, repair and construction of underwater pipelines and structures related to the offshore oil and gas industries. Most ROVs are tethered to the surface vessel with an electrical cable through which control is maintained, and thus are electro-mechanical devices. The vehicles are generally maneuverable in three dimensions (although some maneuver only on the sea floor) through the use of propeller thrusters, and are equipped with various sensors to navigate and to interpret the underwater environment. They are further equipped with manipulators to perform those tasks beyond simple observation and inspection. Figure 1 shows a typical ROV fitted with a manipulator arm.

ROVs provide advantages over divers in many ways including: greater depth capability, greater operating endurance, no risk to human lives, greater range from support vessel, better location sensors (sonar and video cameras), and less cost (except for shallow work). Yet ROVs are inferior to divers in two crucial capacities: the ability to identify objects and their features in the often limited underwater visibilty and the ability to

perform complex tasks requiring manipulation of tools. Both of these abilities depend on what is termed the haptic sense, essentially the information gained through touch and the position of the body and limbs. Consider, for example, the typical task of rigging a lifting sling around a sunken aircraft, something a trained diver can do even in near zero visibility. This task might require the diver to guide a wire strap under the aircraft, while avoiding entanglement of himself or the sling with aircraft wreckage, and to connect the sling with a shackle to a lifting line. In poor visibility he must do much of this by feel, maintaining his orientation, detecting hazards on which he might become entangled, and sensing exactly where the sling and his tools are in relation to each other. This is not an easy task for a diver, but for a ROV it is nearly impossible. All work of this sort by a ROV requires the surface operator to be able to see the manipulator and the work so that the appropriate tool can be properly guided, and even then the process is clumsy at best. Special tools have been devised which allow ROVs to fasten shackles or to grasp particular objects, but visibility is usually a requirement. As another example consider that the simple task of aligning a screwdriver with the head of a screw at an arbitrary angle can be easily performed by a person with eyes closed, but for a remotely operated machine lacking visual feedback it is impossible.

Current ROVs are equipped with sensors which have no counterpart in human senses, such as sonars for navigation, collision avoidance and location of objects. They carry video cameras which often exceed the capabilities of human vision in conditions of poor visibility. However, they lack significant feedback comparable to the human haptic sense. Improvement in this area could vastly increase the utility of ROVs in complex underwater work and is critical to the concept of telepresence.

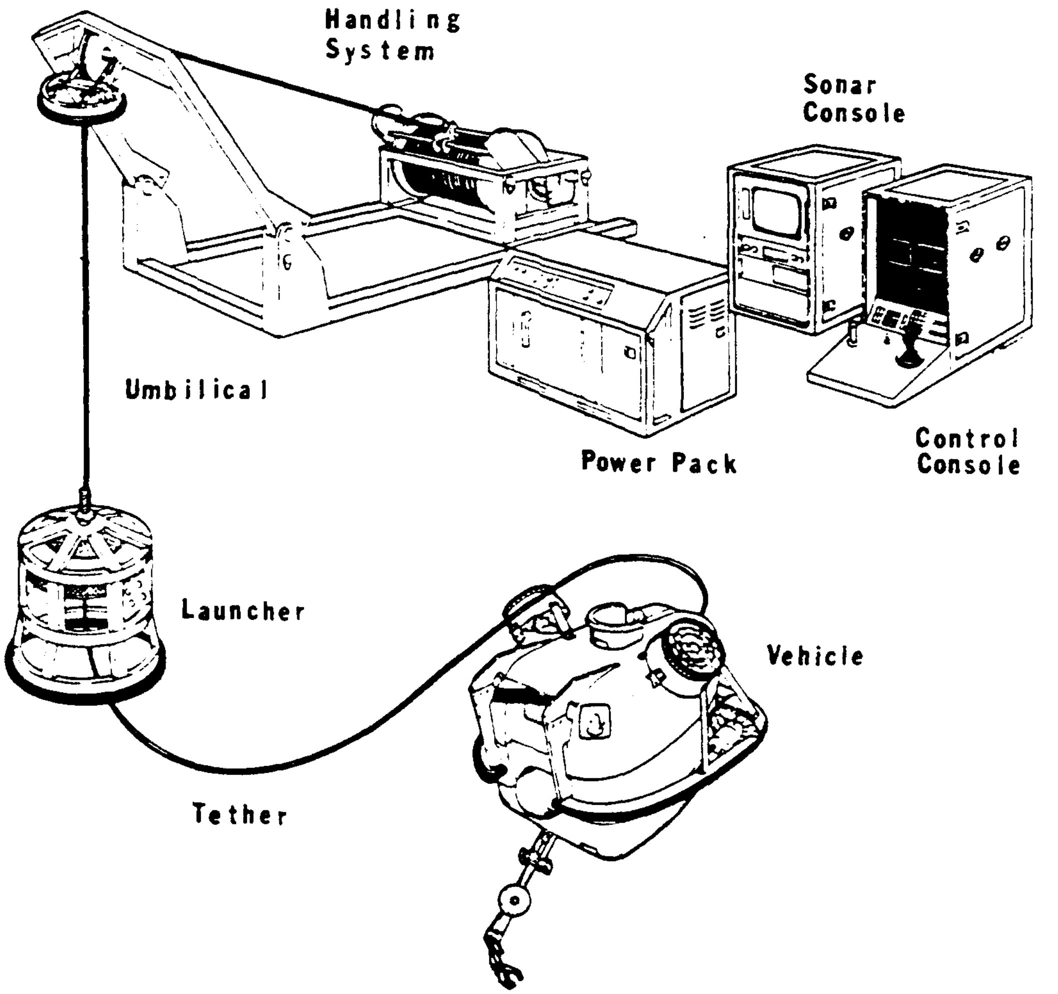


Figure 1. Remotely Operated Vehicle System Components [ROV 84:3]

2. Telepresence

The idea of telepresence is that the operator of a remote robot manipulator (telemanipulator or teleoperator), such as that fitted on a ROV, is provided all the appropriate sensory information needed to make him feel as if he were actually present at the remote location. In the case of an underwater ROV, the operator would ideally see, hear and feel exactly as if he were present on the ocean bottom but without the distracting and potentially harmful environmental effects. He would be able to sense the position and orientation of the manipulator and operate it as if it was his own hand. One concept of the next generation of ROVs, illustrated in Figure 2, is TOPS, an acronym for teleoperator/telepresence [Pepper 86]. In order to achieve telepresence, advances must be made in several areas:

a. Visual and Auditory Sensing

Modern underwater color video cameras perform quite well as a substitute for human vision and, by using stereo cameras, depth perception can be provided. However, in human vision, head movement is an important component in collection of visual data. Telepresence depends not only on delivering all the right data to the operator but providing it in a natural way. Thus camera movement and display must be tied to head movement.

Auditory sensation is probably the simplest to provide, requiring only a hydrophone to receive the sound of a tool tapping, for example, and earphones to provide the sound to the operator. In air, the bilateral arrangement of human ears allows sound direction to be roughly determined by comparing the difference between the time the sound reaches each ear. Because of the greater speed of sound in water, this ability is lost. However, it is possible to conceive of a signal processor which amplifies the time delay between sound received at stereo hydrophones so that the ears hear as if the sound delay

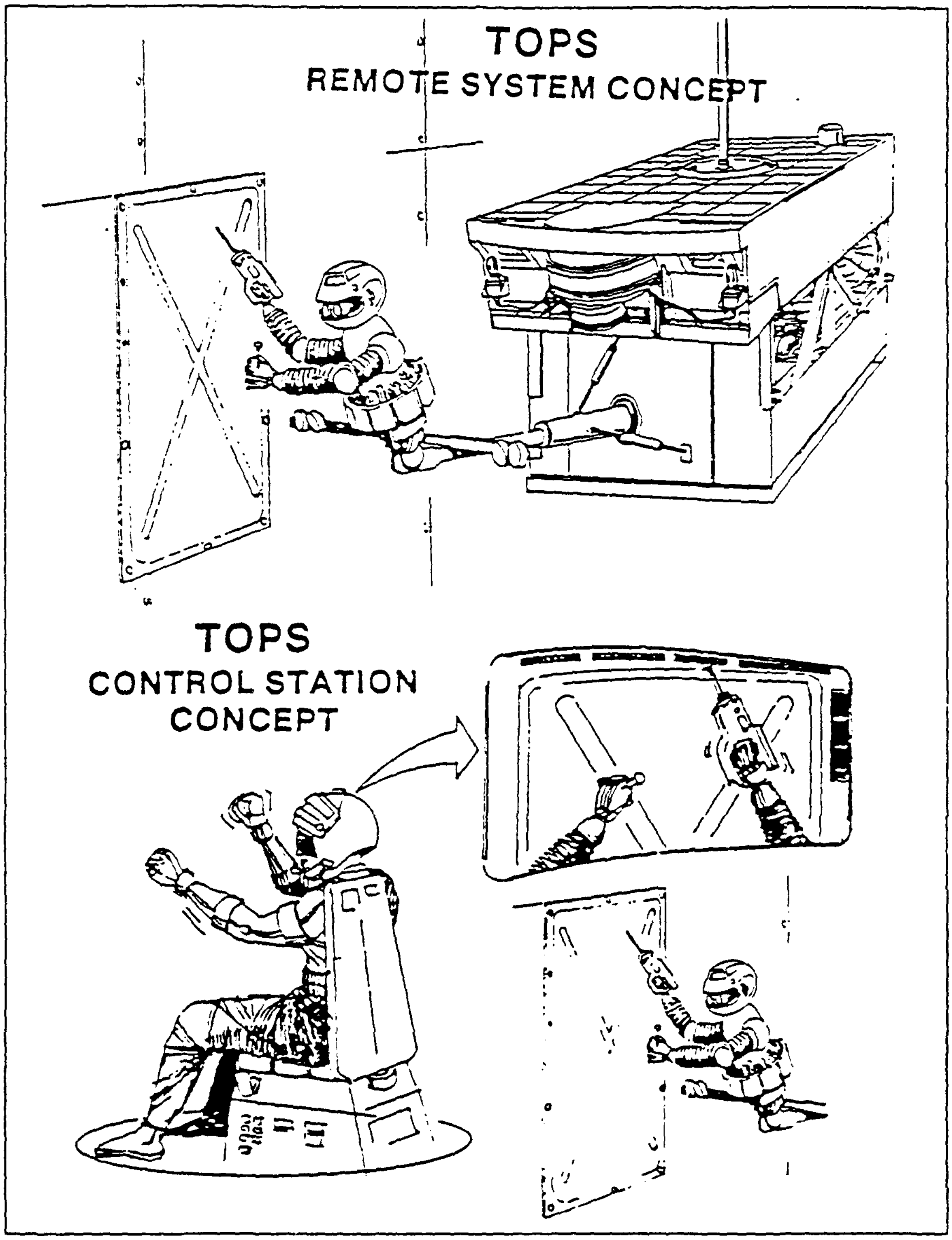


Figure 2. TOPS Teleoperator/Telepresence Concept [Pepper 86]

was in air. With the hydrophones paired with the stereo cameras, the operator would be able tell the direction from which a sound comes thus actually improving on a diver's capability while providing the sensation in a natural way.

ROVs could also be provided with high resolution imaging sonars which could serve as a substitute for vision under conditions of poor visibility. Imaging sonars use high frequency sound (around 500 kHz) to produce an image analogous to television, with sound waves substituted for light. Poor resolution is the major disadvantage of this type of system . [Johnsen 71:159]

b. Haptic Sensing

It is haptic sensing, the gaining of information through touch and the position of the body and limbs, that makes divers so much better at many underwater tasks. Future generations of ROVs must be designed to collect the same haptic information, and deliver it to the operator in a natural way, if telepresence is to be achieved. Driels and Spain found teleoperator performance to depend on four "haptic variables": (1) tactile sensing, (2) tactile display, (3) force reflectance and (4) end effector dexterity. Figure 3 shows the theoretical range of each of these variables and illustrates that telepresence is approached as these variables increase in complexity to approximate human capabilities. Tactile sensing could range from a simple point contact sensor to an artificial skin composed of pressure sensor arrays. A simple disparate mode display would be a light which indicates point contact, while a correspondent mode display would provide pressure stimulus to the area of the operator's skin corresponding to the pressure acting on the teleoperator's artifical skin in contact with an object. Terminus force reflectance provides feedback only of the forces felt at the end of the telemanipulator, while an anthropomorphic arrangement would also transmit forces, which depend on the orientation of the telemanipulator, to the joints of the operator's arm and hand. Finally, end effector dexterity

can vary from a minimum for a single probe to the goal of a multiple-finger, dextrous artifical hand. [Driels 90:873]

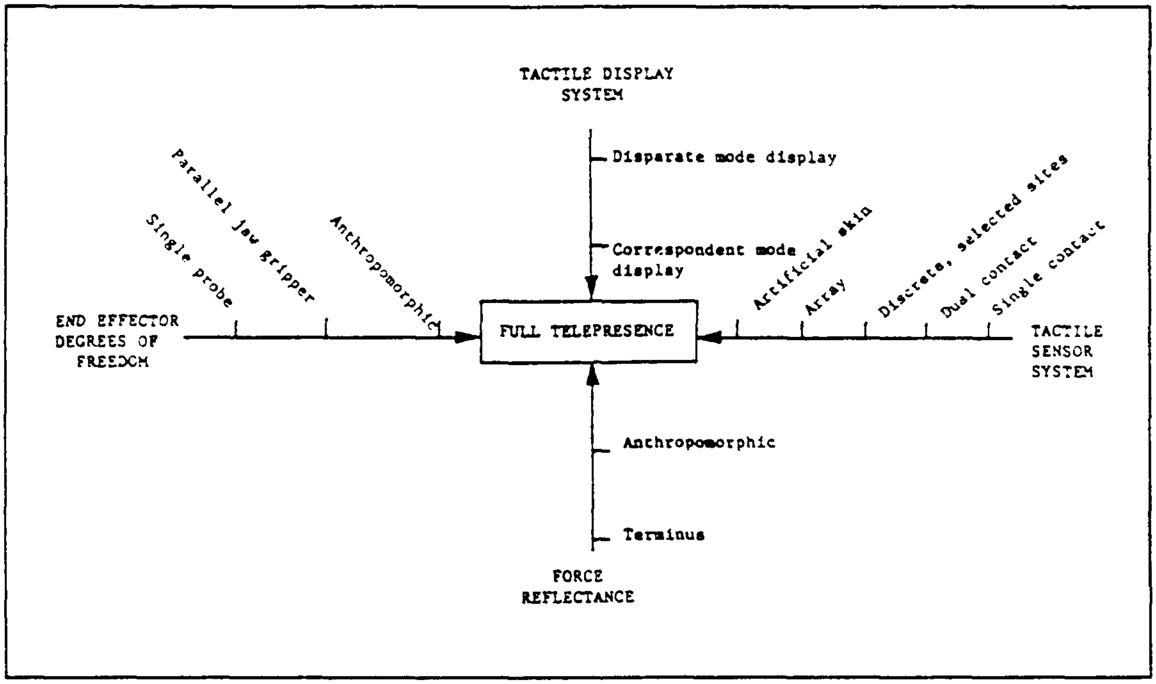


Figure 3. Haptic Variables [Driels 90:873]

Current telemanipulator technology can provide simple force feedback from the end effector and, in a few cases, provide force feedback at the joints of the arm through an exoskeletal arrangement. Touch sensors which indicate contact with a surface have also been incorporated in some designs. However, even these forms of haptic feedback are not available on current ROVs.

B. THE HAPTIC SYSTEM

1. Definition

Haptics was originally applied by Gibson to describe "the perceptual system by which animals and men are *literally* in touch with the environment" [Gibson 66:97]. Klatzky and Lederman describe it as the perceptual system incorporating all of the sensory

inputs derived from involvement of skin, muscles, and joints [Klatzky 87]. This includes a tactile system derived from cutaneous sensors of pressure, vibration, temperature and pain, and a kinesthetic system which senses position and movement through receptors in the muscles and joints. Driels and Spain go further in defining the haptic system as comprising six main components:

a. Tactile (localized) sensing of fine features.

b. Proprioceptive (kinesthetic) sensing of coarse position.

c. Other sensing systems such as temperature and pain.

d. A two-way communication channel between the central nervous system and the brain.

e. Perception processes to formulate hypotheses about the environment.

f. Motor control mechanisms to re-distribute the primary sensor systems. [Driels 90:872]

Of most significance to this study are the perception processes (discussed in Chapter II) and the proprioceptive sense.

2. Proprioception (Kinesthesis)

The familiar accounting of the five senses leaves out what has been known alternatively as the muscle sense, kinesthesis or proprioception. The ability to detect forces and judge the position and orientation of the various limbs of the body, without looking, comes from this sense. "Kinesthetic stimuli displace or deform the tissue underlying the skin: the connective tissue, bones, tendons, and the capsules of joints." [Wolsk 67:133]

Free or bare nerve endings and several different types of encapsulated nerve endings called mechanoreceptors, located in the tissues described above, detect the stimuli

and convert the mechanical displacement to an electrical signal which is transmitted to the central nervous system. The same kind of mechanoreceptors located in the skin, nail beds and hair folicles provide the tactile sense. Some mechanoreceptors rapidly adapt to mechanical displacement and stop signalling as soon as the motion stops and thus are best suited for detecting transient motion. Others adapt slowly and are therefore able to sense the steady position of a joint.

The sensitivity to kinesthetic stimuli is quite remarkable with experiments having shown that a typical mechanoreceptor called a Pacinian corpuscle can detect movement of as little as .0001 inch [Wolsk 67:135]. Other experiments have shown that joint displacements of less than one half degree could be perceived [Clark 86:13-6].

II. THEORY

A. RECOGNITION THEORIES

Theories of human perception and specifically of object recognition have occupied psychologists for many years. Yet because of the awesome complexity of the human mind and the difficulty of deciphering its workings, there is still considerable controversy and no single accepted theory. Most theories are based on experimental observation of visual perception, as this is normally the dominant human sense in interpreting the environment. Most learning and recognition theories agree that sensory stimuli produce an image (not necessarily visual) of an object which is compared to an internal representation stored in memory. Where they differ is in how the sensory and internal images are composed and how they are compared in the recognition process.

1. Gestalt

The basis of the Gestalt theory of perception is that an object is perceived as a whole which is greater than the sum of its parts [Weintraub 68:4]. Thus an object is recognized in a single step procedure in which sensory information is processed in parallel to arrive at a match with the mind's internal representation of the object. This theory specifically excludes the idea that recognition could be a step-by-step process in which features are serially matched with the internal model. In recognition of simple or very familiar objects there is convincing evidence to support this hypothesis [Stark 71:36].

2. Scan Path

Supporting a serial process are two observable phenomena. Experiments have shown that subjects, who first memorize an abstract "target object" and then are asked to recognize the object in a field of several similar objects, take longer to recognize a target

object than to reject a non-target object. Secondly, more complicated objects take longer to recognize than simple ones. Both of these results point to a process by which features are serially matched with an internal model. To match all the features would naturally take longer than to reject an object at the first mismatch, and it takes more time to check more features. The scan path theory proposed by Noton and Stark is such a serial process [Stark 71:34] .

In the human eye the greatest concentration of photoreceptors is located in a small region of the retina known as the fovea. It is only from this region, which represents a tiny fraction of the field of view, that detailed visual information can be obtained. In order to view an object that subtends a larger angle than the fovea, the eye scans the object in a series of fixations on features of interest, interrupted by very rapid movements called saccades. Study of these saccades and fixations, in subjects viewing line drawings or pictures, showed that fixations occurred primarily at angles or other informative details and that definite patterns, termed scan paths, were repeated each time a subject viewed the same picture. Noton and Stark concluded that: "in the internal representation or memory of the picture the features are linked together in sequence by the memory of the eye movement required to look from one feature to the next." [Stark 71:38]

The scan path and its internal representation, the "feature ring", is graphically illustrated in Figure 4 [Stark 81:194].

3. Haptic Perception

Most studies of haptic recognition assume that haptic sensory information is essentially translated into a visual image which is then processed in the same manner as visual stimuli. This is what has been called an image-mediated model. This assumption is called into doubt [Klatzky 87:12] because it fails to explain why haptics appear to be so much better in identifying real three-dimensional objects than in identifying raised two-

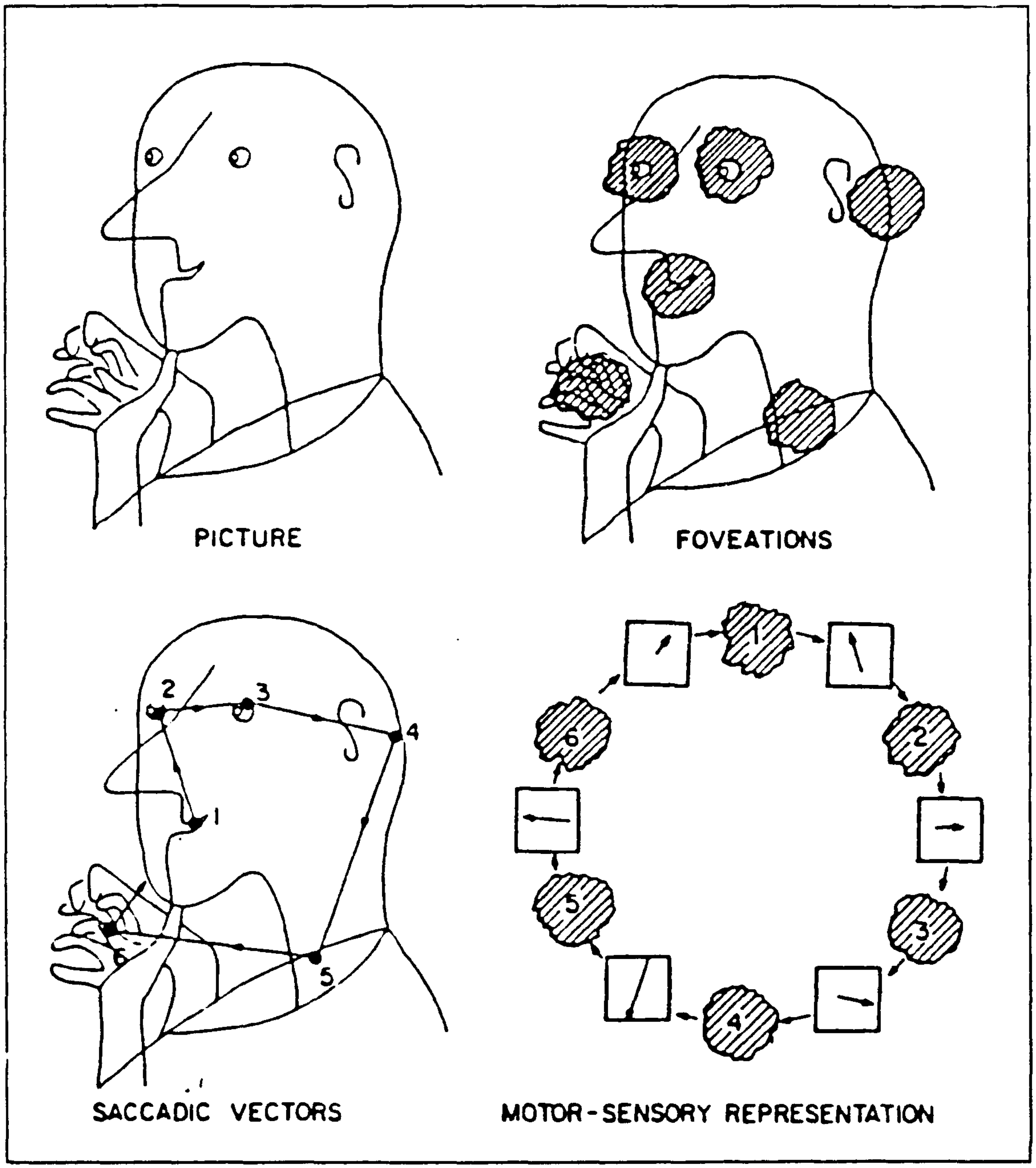


Figure 4. Feature Ring [Stark 81:194]

dimensional line drawings which are essentially profiles of visual images. Klatzky and Lederman contend that there is a parallel haptic processor which is independent of the visual image.

4. Sequential Haptic Probing

Recent research by Acosta [Acosta 91] used a force-reflecting telemanipulator fitted with a single probe, which effectively eliminated the tactile sense from the haptic sytem, leaving only the proprioceptive sense available to the operator. Figure 5 illustrates the experimental set-up. Operators with vision and hearing masked were tasked with identifying a raised two-dimensional letter of the alphabet on a remote task board using the telemanipulator. Subjects invariably followed the contour of the letter, identifying features in sequence until the entire character became recognizable. Significantly, but not suprisingly, there were no saccadic movements from one side of the letter to the other as would be expected in vision. Without the benefit of peripheral vision to guide movement of the probe, the subject would try to maintain contact with the letter being investigated and thus had no choice but to examine each feature in the order encountered.

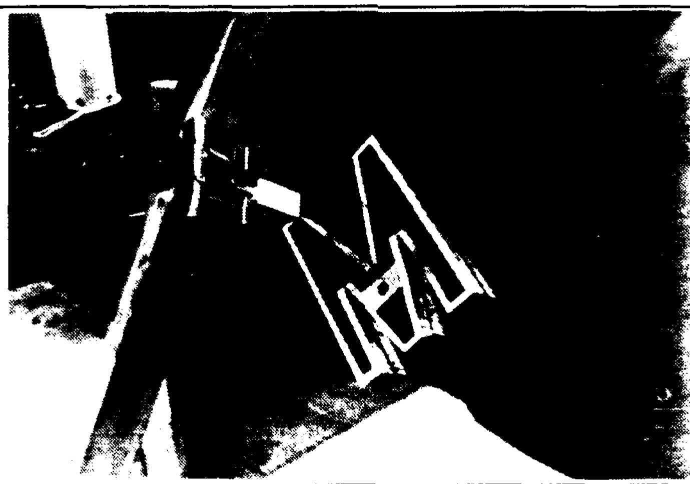


Figure 5. Telemanipulator Probing Object

B. HAPTIC RECOGNITION HYPOTHESIS (FINITE MEMORY MODEL)

It is proposed that for the type of haptic probing just described, in which only the proprioceptive sense is available, identification or recognition is made by matching the sequence of features encountered with the internal representations of the set of possible objects. Although the question of how that internal representation is stored in memory is beyond the scope of this study, certain variables are believed to directly affect the matching process. These variables are believed to be: (1) ability to correctly identify individual features, (2) ability to correctly interpret spatial relationships between features, and (3) ability to retain a short-term memory of the features, sequence, and spatial relationships. It is further proposed in this model that the process in the human brain is comparable to searching the memory library for the best match, perhaps accomodating a limited number of incorrectly identified features if no confusion with other possible matches occurs. It is the limited short-term memory aspect from which the name Finite Memory Model is derived. Figure 6 illustrates in a block diagram how the Finite Memory Model combines these variables to arrive at overall effectiveness of object recognition.

1. Feature Identification

In viewing a simple line drawing visually, it was found that the features which tend to attract the foveal fixations were angles or points of maximum curvature. Figure 7 shows that by selecting points of maximum curvature and connecting them with straight lines all the essential information needed to recognize an object, as complicated as a sleeping cat, is retained [Stark 71:37]. It is believed that the same type of features are most significant in this type of proprioceptive haptic probing. Correct identification of specific features appears to depend on both physiological and mechanical effects.

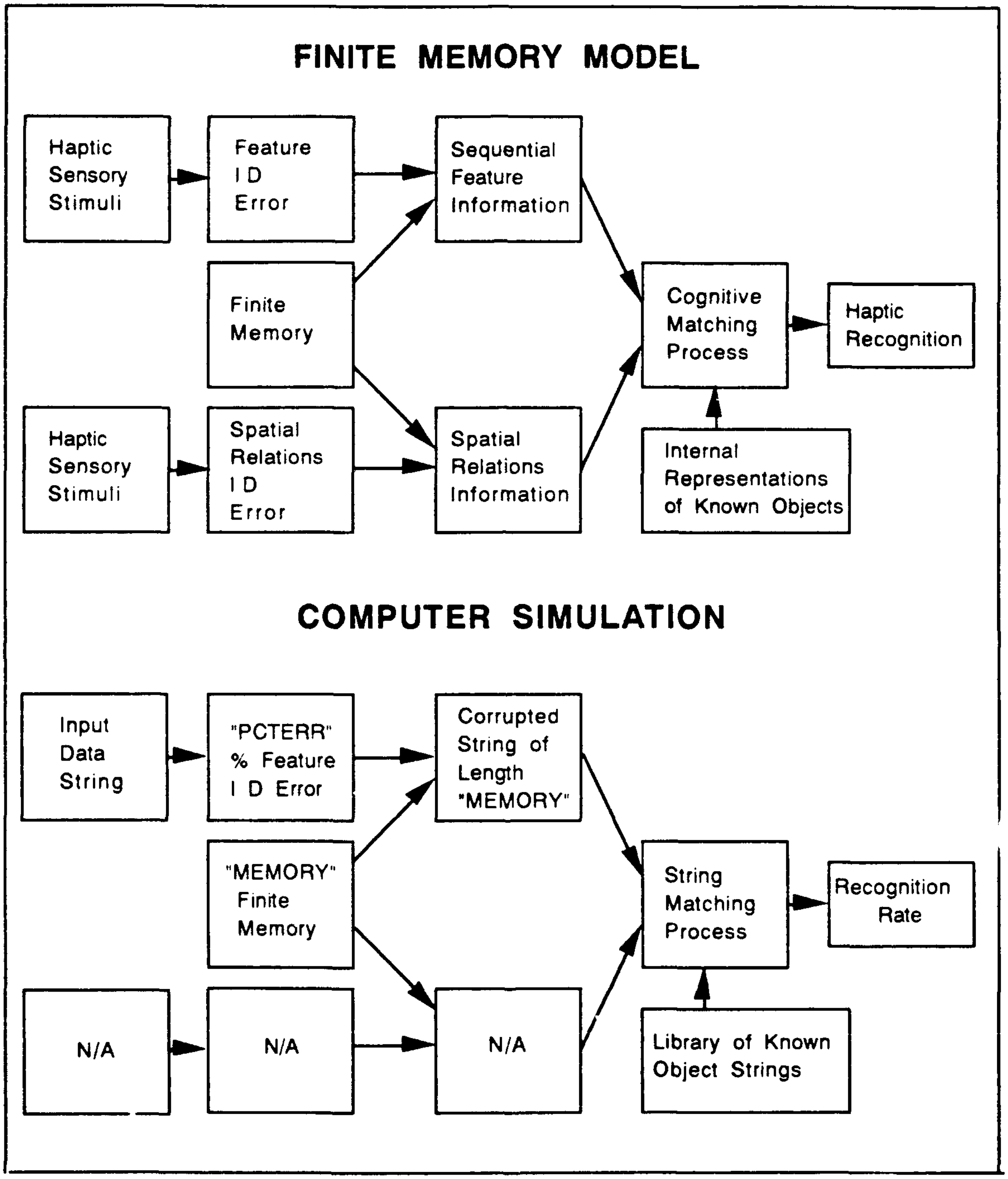


Figure 6. Block Diagrams of Theoretical Finite Memory Model and the Computer Simulation

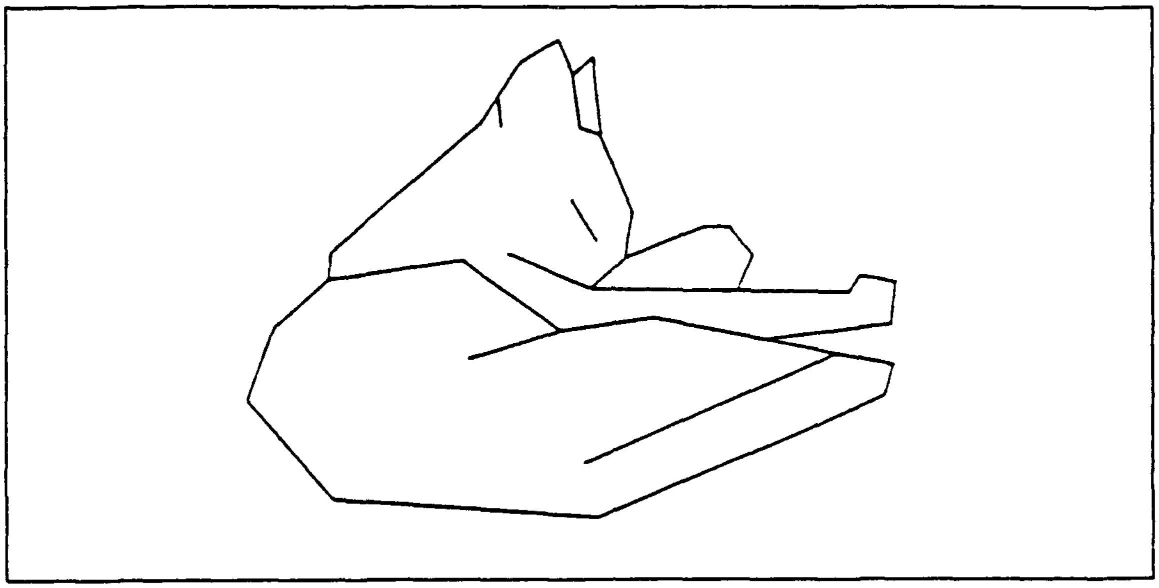


Figure 7. Sleeping Cat [Stark 71:37]

a. Physiological Effects

The effects which measure the difficulty of identifying a raised two-dimensional object with a hand-held probe rather than a telemanipulator are primarily physiological. They derive from the accuracy of the proprioceptive sense in detecting motion and force, and the ability to exercise fine motor control of the arm and hand to direct the investigation of the object. Driels and Spain found that subjects using such a hand-held probe were able to trace raised letters of the alphabet and identify them with much greater ease than when using the same probe mounted at the remote end of a telemanipulator [Driels 90:877].

b. Mechanical Effects

Why was it more difficult to identify objects and the features they comprise with the telemanipulator? The answer lies in the weaknesses of the mechanical telemanipulator in duplicating the function of human limbs. Several effects detract from its ability to perform as smoothly and efficiently as the human arm and hand.

(1) Friction. When motion of the probe is impeded by friction between the probe and the object, between the probe and the task board on which the object is placed, or within the telemanipulator itself, the effect is analogous to noise in an electrical measurement circuit. It becomes more and more difficult to distinguish the constraint forces imposed by the contours of the object from the frictional forces. Operators using a telemanipulator with an aluminum probe to investigate wooden letters on an aluminum taskboard could often not distinguish when they had lost contact with the letter because of the friction between probe and taskboard.

> the probe appeared to be following an edge quite accurately, but when a corner occurred, the probe continued to move in a straight line along the taskboard, constrained only by the end-point frictional forces. The operator was unaware of the situation until the length of the perceived edge became larger than his *a priori* expectations. [Driels 90:876]

The experiment described above resulted in less than 40 percent success rate in object recognition. Using the same telemanipulator but by substituting a plastic probe and a mylar covered taskboard to reduce friction, near perfect success rates and much shorter recognition times have been achieved [Acosta 91].

(2) Stiffness. If the remote end of a telemanipulator is not absolutely stiff, when it contacts a solid object the effect at the operator will be of a constraint force which builds over time to a steady state value. Thus the remote probe behaves with less stiffness than would a human hand in its place. This will certainly complicate the investigation process which appears, based on this study and previous work [Driels 90], to depend at least in part on the rate at which forces are detected. Clearly, in the extreme, one could not expect to be successful using a probe made of rubber. Although not apparently a problem in the purely mechanical telemanipulator used in the laboratory, insufficient stiffness could be quite significant in electromechanical or electrohydraulic units

in ROV applications. The desired level of stiffness may require excessively high control gains.

(3) End Effector Size. Clearly the size of the probe must be small in comparison to the detail of the features that must be extracted for their identification.

(4) Inertia. The mass of the telemanipulator is generally much greater than that of the human arm with the result that it can not be as easily accelerated, which in turn means that the object can not be traced as quickly. In addition, contact is lost when sharp corners are encountered because of the greater inertia. The recognition process is thus hampered by the slow data collection rate, which places increased demands on the memory and makes spatial orientation more difficult, and by the need to regain lost contact at sharp corners.

(5) Compliance. Compliance is defined as "the match between the manipulatory requirement of the task and the motion capabilities of the teloperator" [Johnsen 71:82]. In the experiment which compared a hand-held probe to a telemanipulator-held probe, a much greater level of compliance was seen with the hand-held probe.

> It was also noticed that the human arm could generate variable compliance in different directions relative to the taskboard. In the exploratory procedures observed, the probe was very compliant normal to the board, which also assisted in reducing the sudden build up of end point frictional forces, yet was stiff in any direction parallel to the board so as to generate a rapid change in contact force if motion other than along the object boundary were to take place. [Driels 90:877]

(6) Kinematic Redundancy. Contributing to the compliance just discussed is the fact that the human arm has very redundant kinematics such that there are many more degrees of freedom than required to position a single probe in a given orientation. For example, the major limbs may be repositioned without changing the

orientation and position of the hand. By contrast, most telemanipulators have little or no kinematic redundancy. Even with comparable redundancy, but without anthropomorphic force feedback to the joints of the operator's arm, a similar level of compliance would be difficult to achieve.

2. Spatial Information

The ability to assess spatial relationships between features depends not on the constraint forces imposed by the object but on the aspect of proprioception which allows one to judge the location of the hand by the position of the various joints of the arm. In addition, the definition of particular features combined with the time or distance between features may also provide spatial information, if in a somewhat encoded form.

a. Repeatability

If a person was blindfolded and asked to move his hand back and forth between two points on a piece of paper located at arm's length, each time making a mark with a pencil, the marks would not all fall on the same two points. A circle drawn around the pencil marks at each point could be described as a repeatability circle, which indicates the accuracy with which the person could locate the point using only the proprioceptive sense. If the spatial separation of features of an object is not significantly larger than this repeatability circle, accurate spatial information will be impossible to obtain. The inference is that spatial information is likely to be more important in identifying large objects, where the spatial relationships are more easily discerned, than in small objects.

b. Spatial Information Inherent in Feature Definition

The features that make up the letters of the alphabet are taken in this study to be angles (acute, right, obtuse, inside, outside), curves of varius radius (large, small, inside, outside), and straight sides of various lengths (long, medium, short). These features by their very definition contain coded information about the spatial relationships of

sequentially adjacent features. For example, an outside acute angle followed by a long straight side points to the next feature as surely as a vector. How important this coded information is to this type of haptic recognition is uncertain but its possible impact cannot be ignored.

c. *Spatial Information vs Sequential Information.*

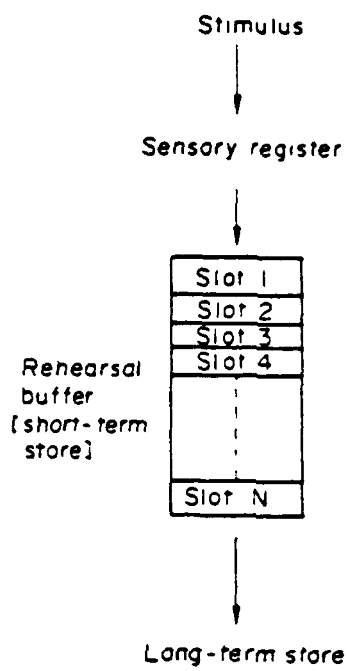
If knowledge of spatial relationships between features were the dominant requirement to arrive at successful identification, one would expect subjects to make excursions with the probe to establish spatial relationships between features which are not sequentially adjacent. However, this is not generally seen because the subject is constrained to maintain contact with the letter or lose the continuity of the sequence so far encountered. If the subject removes the probe from contact with the letter he can not, with confidence, return to the same point and continue his search. A notable exception is the case of the letters "W" and "M" which, with arbitrary orientation, are very difficult to distinguish by haptic probing of this sort, because the angles of the corners of the letter "W" are close enough to right angles to cause confusion with the letter "M". Only after subjects found that sequential probing of features did not provide sufficient information for identification did they depart from the sequential pattern. At that point, the probe would be moved directly from one side of the letter to the opposite side in an attempt to detect whether the opposite sides were parallel (a spatial relationship), as in the "M", or not, as in the "W" [Acosta 91:107]. The initial conclusion is that the information obtained by the sequence of features is more important and more easily obtained than spatial information.

3. Finite Memory

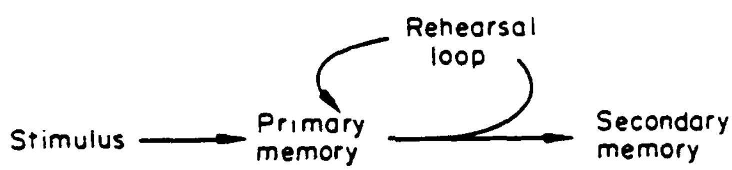
Human memory is commonly divided into three categories: immediate (sensory) memory, primary memory, and secondary memory, although there is no consensus on what these terms mean precisely. Immediate memory is measured in

milliseconds and is the time during which raw sensory stimuli can be retained for processing into semantic form. The primary memory has been estimated by various researchers to be limited to about seven items. It has been described as a buffer store in which a limited number of slots are available. Figure 8 illustrates two such theoretical models of memory. The longer an item remains in the buffer, the more opportunity for rehearsal and the more information about it is transferred to secondary memory. When a new item enters the buffer, one of the old items is displaced. Secondary memory is more permanent storage. Items can be retrieved if they are located in either primary or secondary memory. The two main effects thought to hamper efforts to retrieve data from memory are decay of unattended data (particularly in primary memory) and interference between similar items. [Underwood 76]

As applied to the Finite Memory Model, each feature will be considered an item that occupies one slot of primary memory. Similarly, the information which describes the spatial relationship between two features (except that which is inherent in the feature definition) would likewise occupy one slot. Aspects of the investigation process which hamper memory would include any action which distracts the attention from concentration on the sequence of features so far encountered. Repetitive probing of a feature which is difficult to identify may not only distract from rehearsal and hence transfer to secondary memory, but may actually require room in the buffer and directly displace other items from primary memory. It should also be noted that the opportunity for confusion among similar features will likely lead to interference when trying to recall the information. On the other hand, efforts to group individual features into larger memorable shapes may improve the ability to remember enough features to recognize the entire object.



The model of memory proposed by Atkinson and Shiffrin. Primary memory here is as rehearsal buffer of limited capacity



The primary and secondary memory system envisaged by Waugh and Norman.

Figure 8. Two Memory Models [Underwood 76:79,55]

III. COMPUTER SIMULATION

A computer simulation using Microsoft© Quick Basic language has been devised in order to approximate the Finite Memory Model. The essential elements of the program include: a library of known objects, a mechanism to simulate errors in feature identification, a mechanism to impose a limit on short-term memory, and a cognitive process which matches the input data to the objects in the library. The intent of the simulation is not to duplicate the proccesses of the human mind but to provide a proccess which accomplishes the same task of recognition and allows the key variables to be adjusted as desired. Comparison of the efficiency of the computer simulation and the human mind under similar conditions will give a measure of how close to reality this model comes. A block diagram of the computer simulation juxtaposed against the theoretical Finite Memory Model is contained in Figure 6.

A. LIBRARY OF OBJECT MODELS

The human mind contains representations, or models, of all known objects against which sensory input is compared in order to achieve recognition. In this simulation, a library of encoded descriptions of objects is maintained against which input data will be compared.

1. Object Set and Standard Features

Any set of two-dimensional shapes could be used in this computer simulation as long as they can be adequately described in code. The english alphabet in block, capital letters was chosen in this instance because it is familiar, it comprises a well-defined set, and it has been used in previous work [Driels 90] and [Acosta 91] . Standard features of the letters and their codes are:

1 = Outside Right Angle

2 = Inside Right Angle

3 = Outside Acute Angle

4 = Inside Acute Angle

5 = Outside Obtuse Angle

6 = Inside Obtuse Angle

7 = Large Outside Curve

8 = Small Outside Curve

9 = Large Inside Curve

0 = Small Inside Curve

L = Long Straight Edge

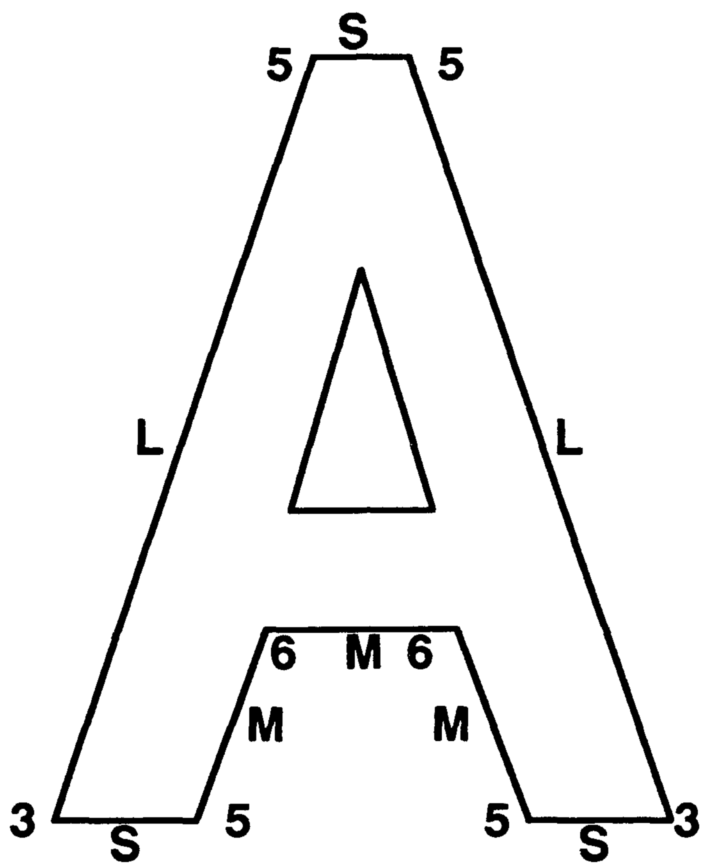
M = Medium Straight Edge

S = Short Straight Edge

2. Character String Descriptors

The encoded representation of the letters in the library consists of a circular character string of the features that make up the letter in sequential order as one traces the letter's contour. The circular aspect, accomplished by repeating the string and connecting it end to end, ensures that the entire contour of the letter is represented regardless of the actual starting point. Examples are provided in Figures 9, 10, and 11. These "library strings" are contained in the subprogram "SUB SEQUENCE" and are translated into familiar letter names by the subprogram "SUB ALPHABET".

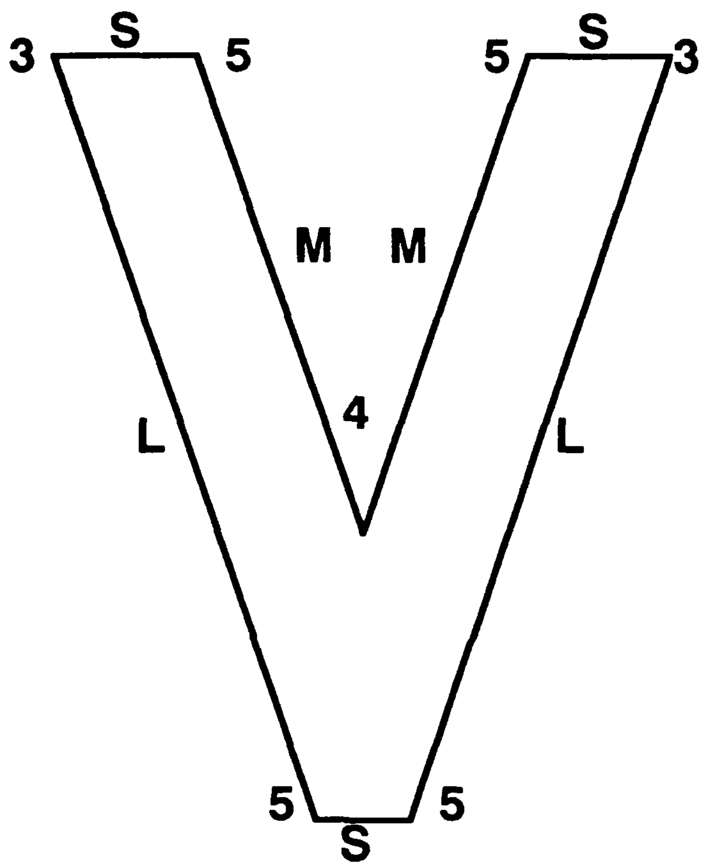
Depending on the direction in which the letter is traced, a different sequence of features will be encountered, unless a letter is symetrical about some axis. To accomodate this fact, unsymetrical letters are listed twice (e.g. F(CW) for clockwise and F(CCW)for counterclockwise). Many letters contain internal pockets which can not be reached without



Character String (Clockwise from Bottom Left):

3L5S5L3S5M6M6M5S3L5S5L3S5M6M6M5S

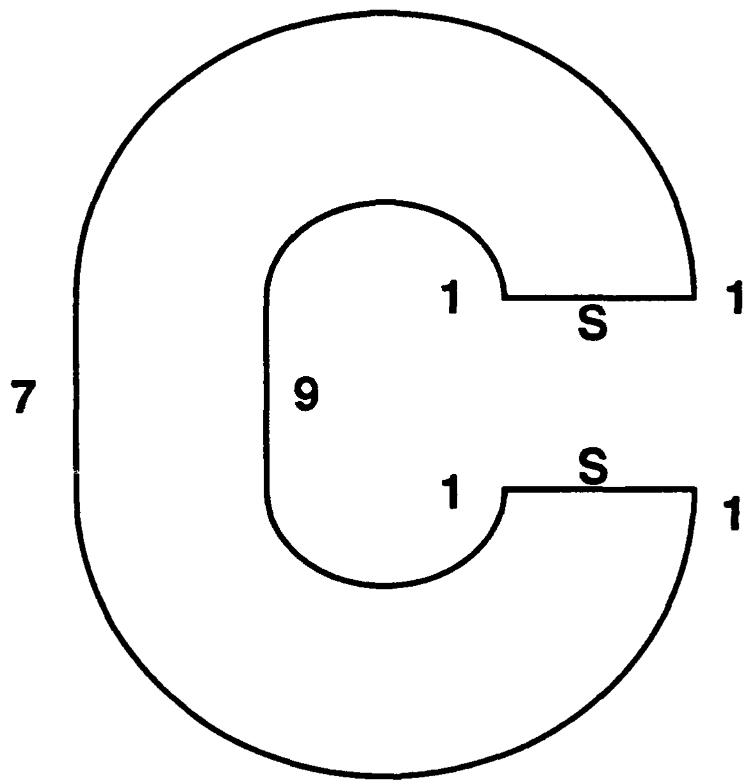
Figure 9. Letter "A"



Character String (Clockwise from Bottom Left):

5L3S5M4M5S3L5S5L3S5M4M5S3L5

Figure 10. Letter "V"



Character String (Clockwise from Bottom Left):

71S191S171S191S

Figures 11. Letter "C"

leaving the outside contour. Initially this break was indicated by a hyphen in the character string. However, because this break may occur at any point in the outside contour and the fact that each letter can be uniquely defined from its outside contour alone, the inside contour was ultimately ignored.

B. FEATURE IDENTIFICATION

The mechanical and physiological effects that lead to errors in feature identification must be simulated by artificially introducing errors into the input data for the computer program. The actual haptic search path of a human subject is simulated by an input data string. The input data string is constructed from the library string encompassing one circuit around a letter starting at a random point.

In order to simulate a given amount of error in identifying features, the basic input data string is corrupted by injecting a specified percentage of incorrect features. The locations of the altered features within the string are chosen randomly. Each altered feature is replaced only with a substitute feature which might reasonably be expected to be confused with the correct feature. Thus, an outside right angle (1) could be replaced by an outside acute angle (3) or an outside obtuse angle (5) but not with a long straight edge (L). Subprogram "SUB FEATURE1" establishes the acceptable substitutions. From this group of possible misidentifications, the substitute feature is chosen randomly. By specifying a desired percentage of error in identification of features, program "INBUILD.BAS" constructs an appropriately corrupted input data string. A series of such corrupted input data strings with different values of feature identification error can be combined for ease of program execution using program "SERINBLD.BAS".

C. SPATIAL INFORMATION

No attempt is made in this program to model the spatial information gained by proprioception independent of the spatial information encoded in feature definitions. If the initial conclusion that such spatial information is less important than that contained in the sequence of features is wrong, then the simulation results should not compare favorably with actual human experience.

D. FINITE MEMORY

A limit on the number of features which may be remembered, and are thus available for matching with the internal representations, is simulated by limiting the length of the input data string which can be used to match with the library strings. An integer value is specified by the program operator as the value of the variable "MEMORY".

E. COGNITIVE PROCESS

1. String Matching

Simulation of the human mind's matching process is achieved by comparing each character in the input data string to each corresponding character in the library string, for all the library strings, until a match is found. For example, consider an input data string that represents the letter "C", shown in Figure 11, started at an arbitrary point on the contour. Its length is eight features. It will be compared feature by feature with every eight feature segment of each library string checking for a perfect match as shown below:

Input Data String: **191S171S**

Library String for Letter "C": 71S191S171S191

[**71S191S1**]71S191S

7[**1S191S17**]1S191S

71[**S191S171**]S191S

71S[**191S171S**]191S Match!

Matching an object with a preconceived set of possibilities, whether visually or with haptics, is made more difficult if all the possibilities are quite similar, such as when the objects all share the same features but only the orientation and sequence distinguishes one from another. Conversely, if objects contain unique features or groups of features, they become more easily recognizable. In visual perception, unique features are immediately apparent, but in haptic investigation of an object, finding unique features depends on where the investigation begins. Take, for example, the letters "A" and "V" illustrated in Figures 9 and 10. Starting at the inside of the left leg of the "A" and tracing in the clockwise direction around the letter to the inside of the right leg, the features encountered would make up the sequence: 5S3L5S5L3S5. Examination of the Figure 10 shows that the letter "V" shares the same sequence, so the two objects can not yet be distinguished. However, tracing in the opposite direction from the same starting point, only four features , 5M6M, are needed to uniquely distinguish the "A" from all other letters in the alphabet. The main point here is that, unlike visual perception, haptic recognition depends on the point at which contact with an object begins.

2. Closest Match

If the input data string has been corrupted by a feature identification error, it will no longer be possible to find a perfect match. In this case, the program conducts the same type of string matching but remembers those library strings for which a close match is found. Close matches are measured by the number of features which do not match. Thus,

the nearest thing to a perfect match is a close match with a single error. In order to qualify as a close match, incorrect features must satisfy the same reasonableness test imposed earlier on the input data strings. They must be similar enough to the correct features in the library string to have been reasonably confused during haptic probing. The subprogram "SUB FEATURE" provides this test of reasonableness.

3. Finite Memory

If, in addition, the memory is limited, the string matching will now take place between segments of the input data strings of length "MEMORY" and segments of the library strings of equal length. Consider, for example, another input data string for the letter "C", this time with a 10 percent feature identification error specified, leading to one error in the eight character string. In addition, the length of the finite memory is set at five features by making "MEMORY"= 5. The matching procedure will now be conducted for each of the four segments of that length:

Input Data String: **S171S191**

Corrupted Input Data String (10% Feature ID Error): **S171S1L1**

[**S171S**]1L1 1st segment (of length "MEMORY" = 5)

S[**171S1**]L1 2nd segment

S1[**71S1L**]1 3rd segment

S17[**1S1L1**] 4th segment

The four segments will be compared with every equal length segment of each library string. Note that in matching with the library string for the letter "C", the 3rd and

4th segments will only yield close matches because of the feature identification error which substituted (L) for (9).

Library String for Letter "C": 71S191S171S191

[**71S19**]1S171S191

7[**1S191**]S171S191

:

:

71S191[**S171S**]191 1st segment Match!

[**71S19**]1S171S191

[**1S191**]S171S191

:

:

71S191S[**171S1**]91 2nd segment Match! etc.

Note that the introduced feature identification error combined with the limited memory may lead to incorrect matches as is the case here for the letter "I" and the 4th segment:

Library String for Letter "I": 1L1S1L1S1L1S1L

[**1L1S1**]L1S1L1S1L

1[**L1S1L**]1S1L1S1L

1L[**1S1L1**]S1L1S1L 4th segment Match!

4. Performance

Matches of segments as described above are divided into two categories: those that would result in correct identification of the letter from which the input string was derived, and those that would result in identification of the wrong letter. A score is computed by taking the ratio of correct matches to the total of correct and incorrect matches. If there are no perfect matches, the same ratio will be taken for close matches of the same number of errors. Thus, if no perfect matches and no matches for only one error were found, the ratio would be between close matches with two errors which would lead to the correct identification, and close matches with two errors which would lead to the wrong identification. Averaging the scores for a set of input strings, containing each letter in the library, will result in a performance measure termed the "recognition rate".

For the example just described, there were two correct matches and a single incorrect match (assuming that no matches were found with library strings other than "C" and "I"):

SCORE = (correct matches)/(correct matches + incorrect matches)

= (2)/(2 + 1)

SCORE = 0.67

The program "MATCH.BAS" is the main program in which the function of string matching takes place. The output is a single value of recognition rate for a given value of feature identification error and a given length memory. Program "SERMATCH.BAS" is a modified version which performs the same function but allows a series of feature identification errors to be used, yielding a series of data points instead of just one. The recognition rate can be plotted against feature identification error to yield a

set of parametric curves for various lengths of memory. All computer programs are included in Appendix A.

IV. EXPERIMENTAL WORK

In order to test the computer simulation of the Finite Memory Model, it is desirable to have reasonable values of the input variables: memory length and feature identification error. The first of these, memory, is independent of the particular telemanipulator in use. Feature identification error, on the other hand, is directly a function of the various mechanical factors described earlier and so must be established separately for each different telemanipulator. Because of the general applicability of the memory length value, it was decided that this would be the focus of the experimental work. Time did not permit conducting feature identification error experiments.

A. FINITE MEMORY EXPERIMENT

The purpose of this experiment was to find a reasonably accurate value for the number of different features, in a particular sequence, that can be remembered by a typical telemanipulator operator. There was no intent to distinguish between the roles of primary and secondary memory in the overall ability to recall a sequence of features. Rather, the purpose was to find the net result of the entire memory process as it might work during actual haptic probing of an object within a time frame that was typical of haptic probing exercises. During this time there would be, of course, considerable rehearsal of previous features in the effort to remember the sequence. But this was expected to be no different than the actual experience during haptic probing of a more complex object, such as a letter of the alphabet. The primary concern in designing this experiment was the elimination of effects which might complicate the task of recalling the sequence of features or which might provide additional information which might aid recall. Two specific effects needed to be eliminated: feature identification error and spatial relationships between features.

1. Eliminating Feature Identification Error

If the subject of this experiment were unable to identify even one of the features with which he was confronted, he would be unable to correctly recall the sequence, regardless of the capacity of his memory. Additionally, if the subject correctly identified all features, but the process of identification was difficult or time-consuming, the effect on measured memory capacity would likely be significant. First, recalling the register analogy for the short term memory, it can be seen that repeated investigation of a single feature could be occupying slots in the register and displacing earlier features. Secondly, decay of unattended data in primary memory is a significant source of memory loss, as described earlier in the theoretical discussion of memory. This decay requires time and distraction, both of which are provided by the effort to identify ambiguous or complicated features. Therefore, features had to be chosen to avoid any likelihood of misidentification.

2. Eliminating Spatial Relationships

Information about the spatial relationship of features could conceivably both help and hinder the memory process. If the spatial information must be remembered independently, and thus occupies slots in the memory register, the number of features remembered might be less. However, if the spatial relationships contribute to forming easily remembered groups of features, the memory process may be facilitated. These effects, plus other perhaps unrecognized effects, made it wise to arrange the sequence of features in such a manner that spatial relationships were insignificant. Essentially, what was desired was a one-dimensional sequence of features.

3. Object / Feature Set

To ensure applicability of the results of this experiment to the object set used in the computer simulation (letters of the alphabet), the identical feature set (various angles, curves and straight edges) was initially considered for this experiment. However, it is

impossible to arrange such elemental features in a one-dimensional sequence. Instead, a set of simple raised two-dimensional geometric shapes, including a square, a 45 degree right triangle, a quarter circle, and a square with a quarter circle removed (essentially the inverse of the quarter circle) was selected. The shapes were all of the same scale, with sides of three inches, and were distinct enough that there would be little chance of misidentification. An empty space in the sequence was treated as an additional feature. The shapes are illustrated in Figure 12. The key assumption in treating these shapes as basic features was that they would be immediately recognizable and memorable not as a group of angles and lines but as a single item. They would thus occupy a single slot in the memory register.

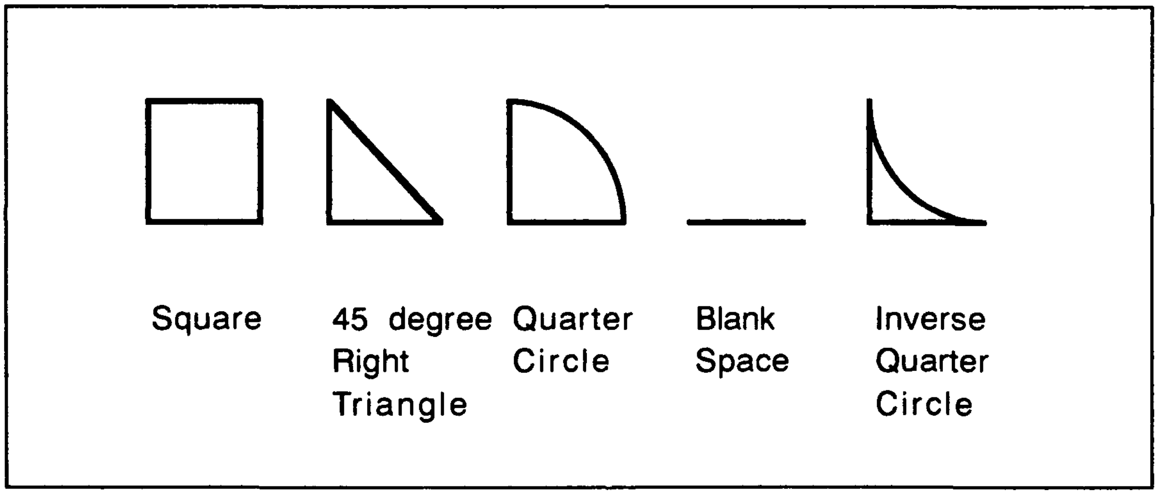


Figure 12 Memory Experiment Features

Another problem, encountered with the first attempts at this experiment, was the tendency of subjects to be very keenly aware of any pattern in the sequence of features and to use the pattern as a memory aid. To combat this tendency, great care was exercised in selecting sequences without patterns and the shapes were placed in either of two orientations: facing left or facing right. Since the subject would now be required to remember two bits of information about each shape (except the square and blank space),

the orientation was treated as a second feature to be recalled. Thus, a shape and its orientation were treated as equivalent to two of the features used to define the letters of the alphabet in the computer model.

4. Experimental Apparatus

The physical apparatus included two main components: a force-reflecting telemanipulator and a task board on which the sequence of features could be arranged. Both are illustrated in Figure 13.

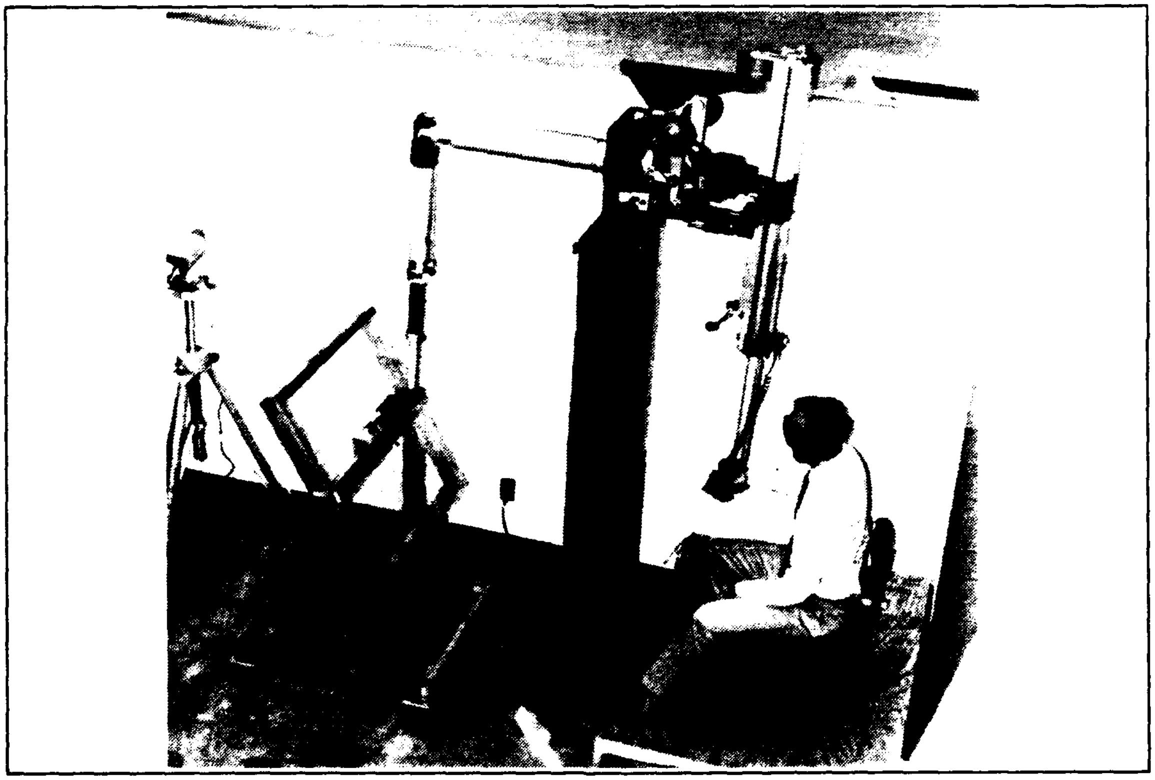


Figure 13. Telemanipulator and Task Board

a. Telemanipulator

A Central Research Laboratories, Mod-8, mechanical master-slave telemanipulator, originally designed for working with radioactive materials, was used in this experiment. It has seven degrees of freedom, one of which is in the parallel gripper

jaws which were fixed in the closed position in order to hold the probe. The probe was a six inch long, 1/4 inch diameter, steel tube fitted with a rounded plastic tip. The telemanipulator is considered to be of the terminus type because it reflects only the forces at the end of the remote arm. A system of antagonistic cables and pulleys allows the operator to sense these forces at the pistol grip handle.

b. Task Board

The task board was a three foot square, plexiglass working surface mounted on a heavy wooden stand, positioned facing the remote end of the telemanipulator,and slightly inclined from vertical, much as an artist would position his easel. A sheet of transparent mylar was fixed to the plexiglass to keep friction between the probe and the board to a minimum. The two-dimensional, raised geometric shapes were cut from 3/4 inch plywood and mounted in the prescribed sequence with a wooden slat clamped onto the task board. The slat was positioned so that the sequence of features appeared as a horizontal line, with only one spatial dimension. Figure 14 shows the probe investigating a sequence of shapes mounted on the task board.

5. Experimental Procedure

Two experienced operators were used as subjects for this experiment. Although the number of subjects was not considered statistically adequate, the results were expected to produce a reasonable figure of memory length for use with the computer simulation. The subject was separated from the task board by a curtain and his vision and hearing were masked. A sequence of features was mounted on the task board and the probe positioned at the starting point. The subject then investigated the sequence with the probe. When he thought that he could remember the sequence, he would stop and draw the sequence on a piece of paper. There was no time limit imposed, but the time used was recorded. If the subject correctly recalled the sequence it would be noted, otherwise his

incorrect drawing was saved. The sequences ranged from six to sixteen features in length and were randomly presented without the subject knowing the length. There was, however, a general trend from shorter to longer sequences, which would allow time for the operator to become accustomed to the procedure with the easier, shorter sequences. Several sequences of each length were presented. Incorrect recall of a sequence was counted as a failure, regardless of how close it was to the correct answer, and a score determined by the percent of correct recall for each sequence length. The value of memory length would be determined from the plot of percent recall versus sequence length.

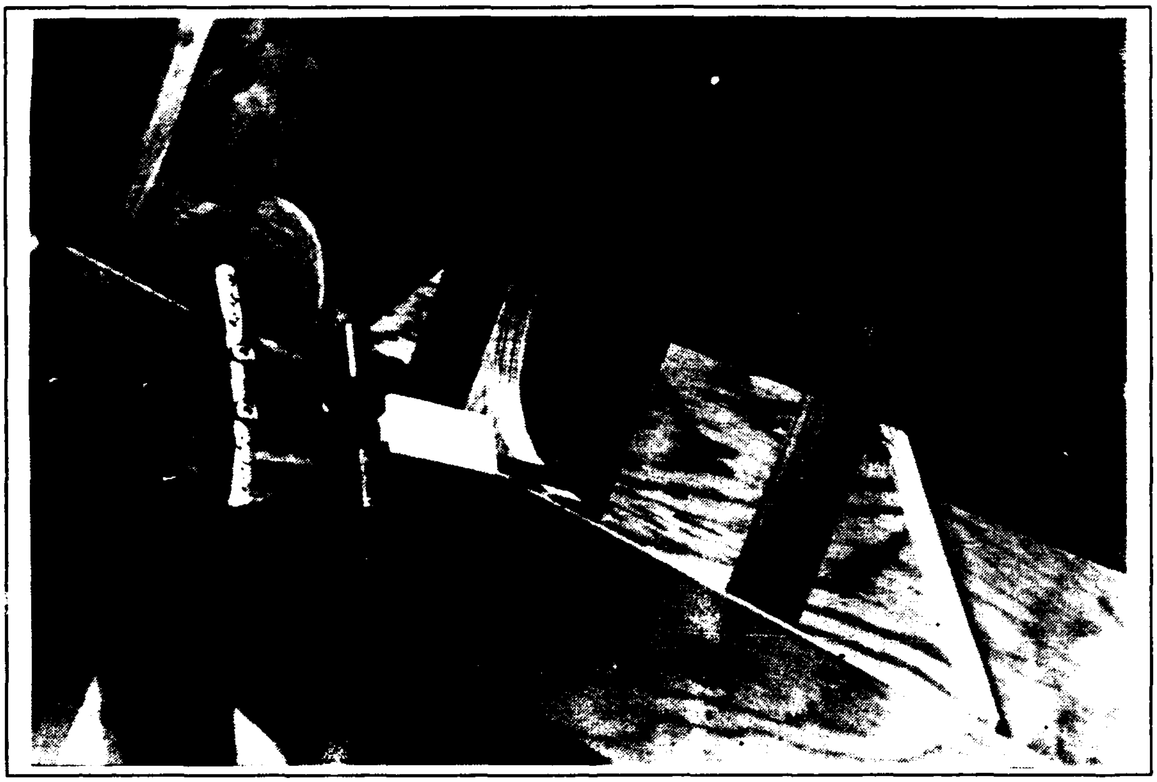


Figure 14. Sequence of Features on Task Board

V. RESULTS

A. FINITE MEMORY EXPERIMENT RESULTS

The two subjects operating the telemanipulator investigated a total of 81 sequences, not including several sequences which were subsequently rejected for obvious patterns which aided recall. Of the total, 23 sequences were incorrectly remembered by the subject. The results, plotted as rate of correct sequence recall versus number of features in the sequence, are shown in Figure 15. The actual data, including the sequences and the subject's recall, is contained in Appendix B. One subject could reliably recall up to 11 features in sequence, after which his recall degraded rapidly. At 14 features he had no successful trials. The second subject experienced similar performance but his threshold was somewhat higher, successfully recalling up to 13 features without error. Sequences with 16 features were always beyond his capacity to recall. Averaging the two cut-off values produces an experimentally derived value for memory length of 12 features.

Feature identification was not difficult and was completed in a single pass of the probe over the feature. Only a single feature in all the trials was felt by a subject to have been misidentified. The subjects, in general, would conduct two passes of the probe over the entire sequence before attempting to draw the sequence. The average time spent investigating a sequence was 10.4 seconds per feature and, as shown in Figure 16, the time per feature was quite consistent regardless of the length of the sequence.

Another significant result is that errors were not randomly distributed within the incorrectly recalled sequences. Examination of the subject's recall of sequences which were not correctly identified shows that when errors were made they were predominantly at

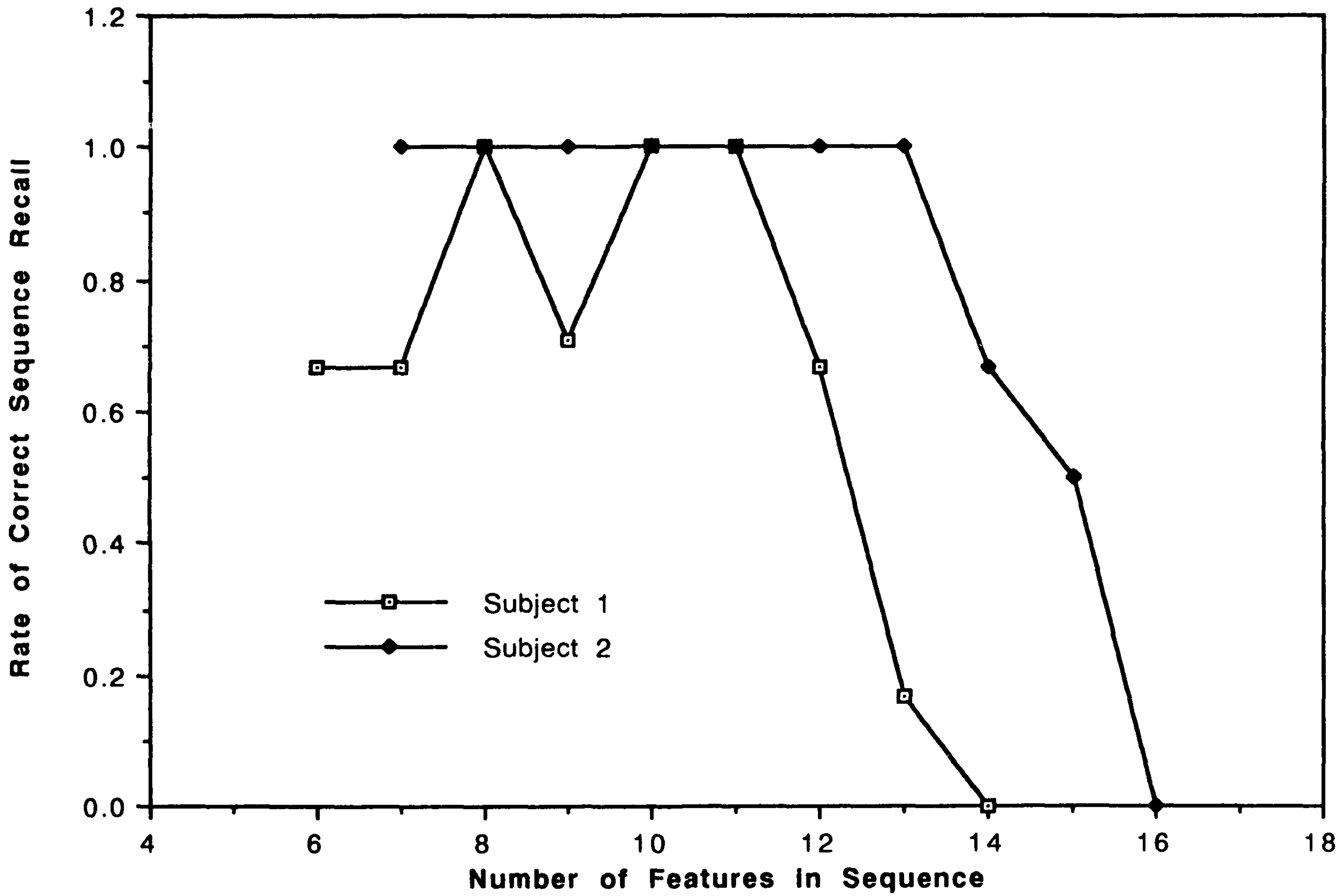


Figure 15. Memory Experiment Results

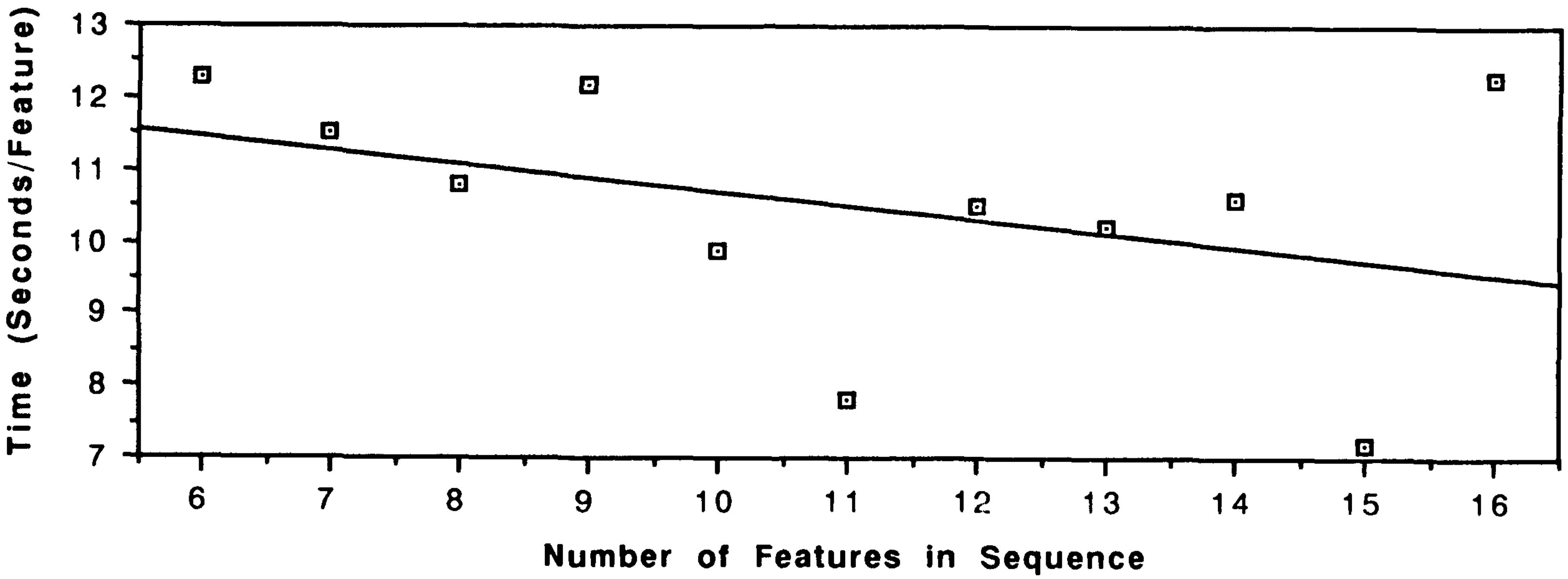


Figure 16. Memory Experiment Elapsed Time

the beginning or end of the sequence. Figure 17 shows with a bar graph the number of errors in the first two and last two shapes as opposed to all other error locations.

B. COMPUTER SIMULATION RESULTS

The computer simulation was run concurrently with the finite memory experiment, which meant that the experimentally derived value for memory length was not available to specify in the input data. Instead, various values of memory length ("MEMORY") were used with the computer simulation to establish a family of curves showing recognition rate versus feature identification error. Once the experimental value of memory length became available, the computer simulation result could be interpolated from the family of curves.

An input data group (created by "INBUILD.BAS") consisted of 33 input data strings, one for each object in the library. It may be recalled that the library contains character string representations of each letter in the alphabet, with unsymetrical letters included twice to account for the difference between clockwise and counterclockwise directions. Fourteen such groups made up an input data series (compiled by "SERINBLD.BAS"). In an input data series, each group had a different feature identification error ("PCTERR") ranging from zero to 90 percent. Running the simulation program ("SERMATCH.BAS") with an input data series resulted in 14 data points from which one curve of the family described above was constructed. Figure 18 shows a family of recognition rate versus feature identification error curves for one input data series. The memory length values used were: 7, 10, 15, and unlimited. Both the raw data points and the best-fit linear curves are shown.

Although these curves show the expected downward trend in recognition rate as more feature identification error occurs, significant departures from that trend at 15 and 30

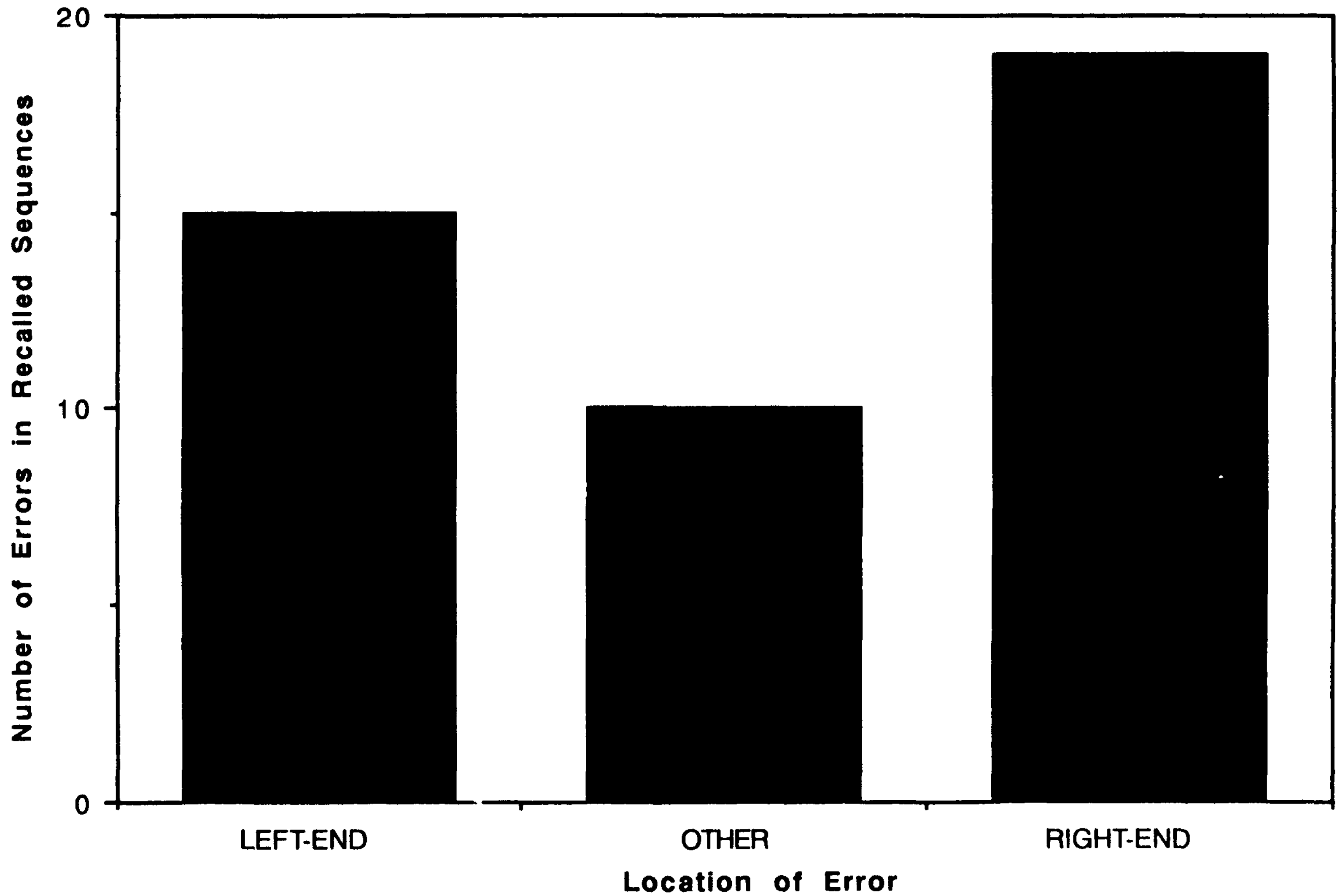


Figure 17. Location of Errors in Recalled Sequences of Features

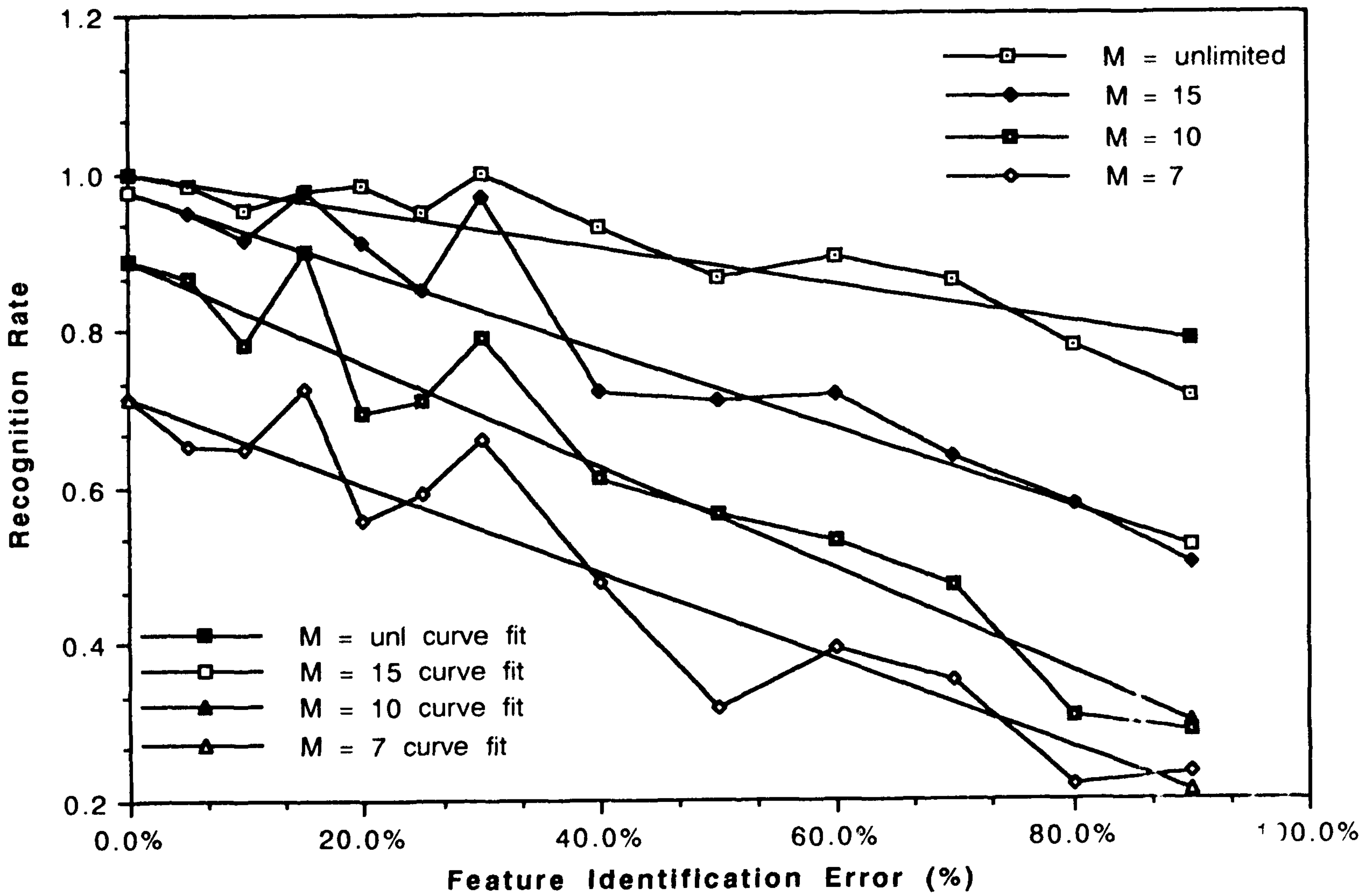


Figure 18. Computer Simulation Results for a Single Input Data Series

percent feature identification error for all values of memory length caused concern. Test runs with several input data groups, all with the same feature identification error and memory length, showed a strong dependence on the input data group and produced a wide range of recognition rates, ostensibly for the same input variables. The reason for this dependence is that in making up the input data group, each input data string is started at a random position in the library string and incorrect features are randomly substituted to give the prescribed feature identification error. This results in radically different groups of input data strings being compared to the library of object strings to find best matches.

To solve this problem, a statistical sample of ten different series of input data groups were prepared. Using a value of ten as the best estimate of memory length, the computer simulation was run for each of the ten input data series. Figure 19 shows the result of statistical analysis of the resulting data. Mean values of recognition rate and the standard deviation are plotted, revealing a curve much closer to linear than was evident in Figure 18. Regression analysis of the mean values gives a correlation factor of 0.993 and a best-fit linear curve equation as shown in the figure.

Time did not allow similar statistical analysis, with multiple input data series, for other values of memory length. However, regression analysis of the "MEMORY=10" data points in Figure 18, which resulted from a single input data series, yields a curve fit very close to the one obtained for the mean values in Figure 19. Given the close similarity in the shape of all the curves in Figure 18, it appeared safe to assume the same to be true for other values of memory length, and it became possible to extrapolate the curve fit for the data in Figure 18 to produce the final family of curves shown in Figure 20. An interpolated curve for the experimentally derived memory length of 12 is also included in this figure. It should be remembered that the desired result was a measure of the recognition performance of the computer simulation that would allow comparison with results of future haptic

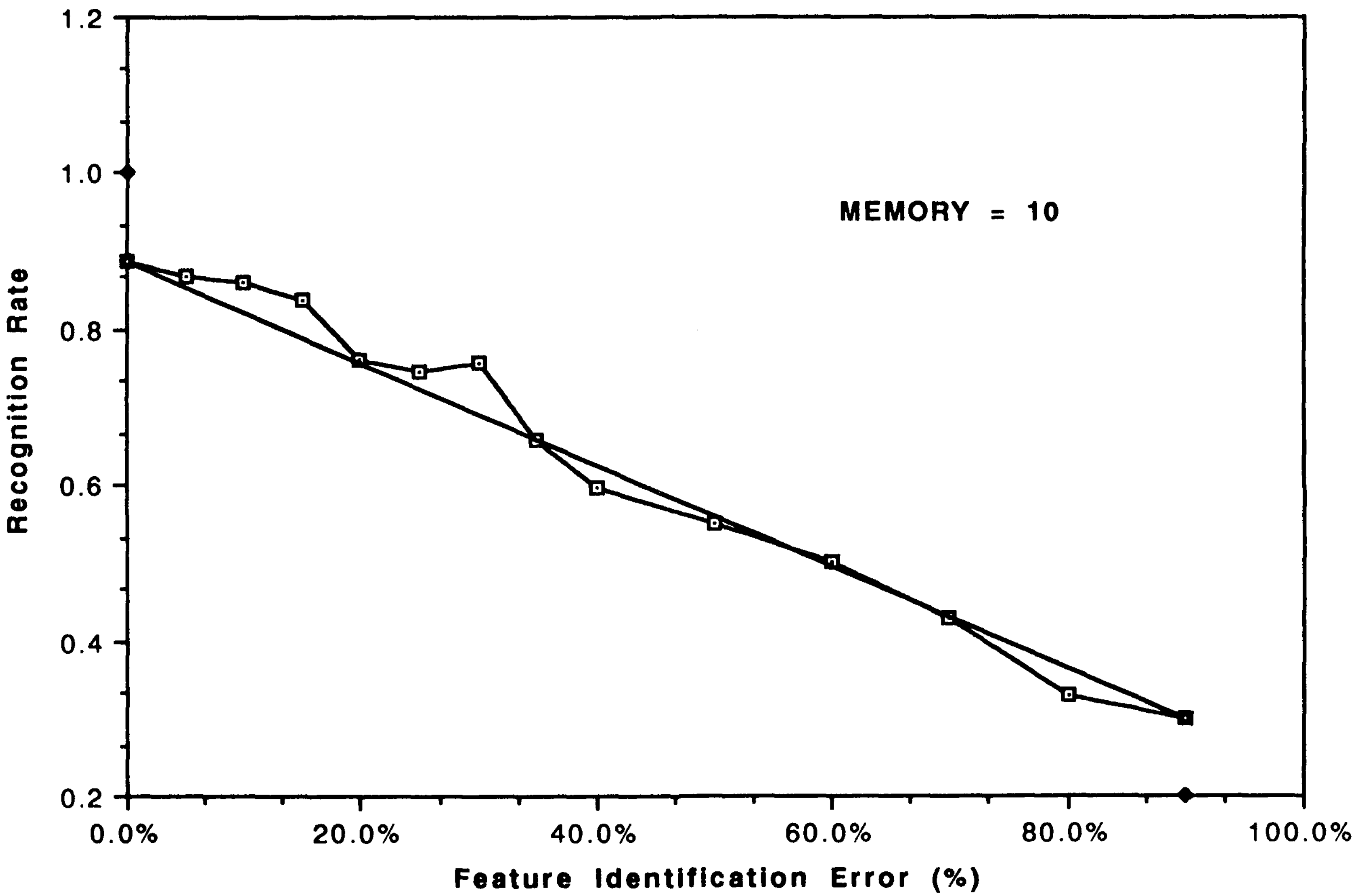


Figure 19. Mean of Computer Simulation Results for 10 Input Data Series

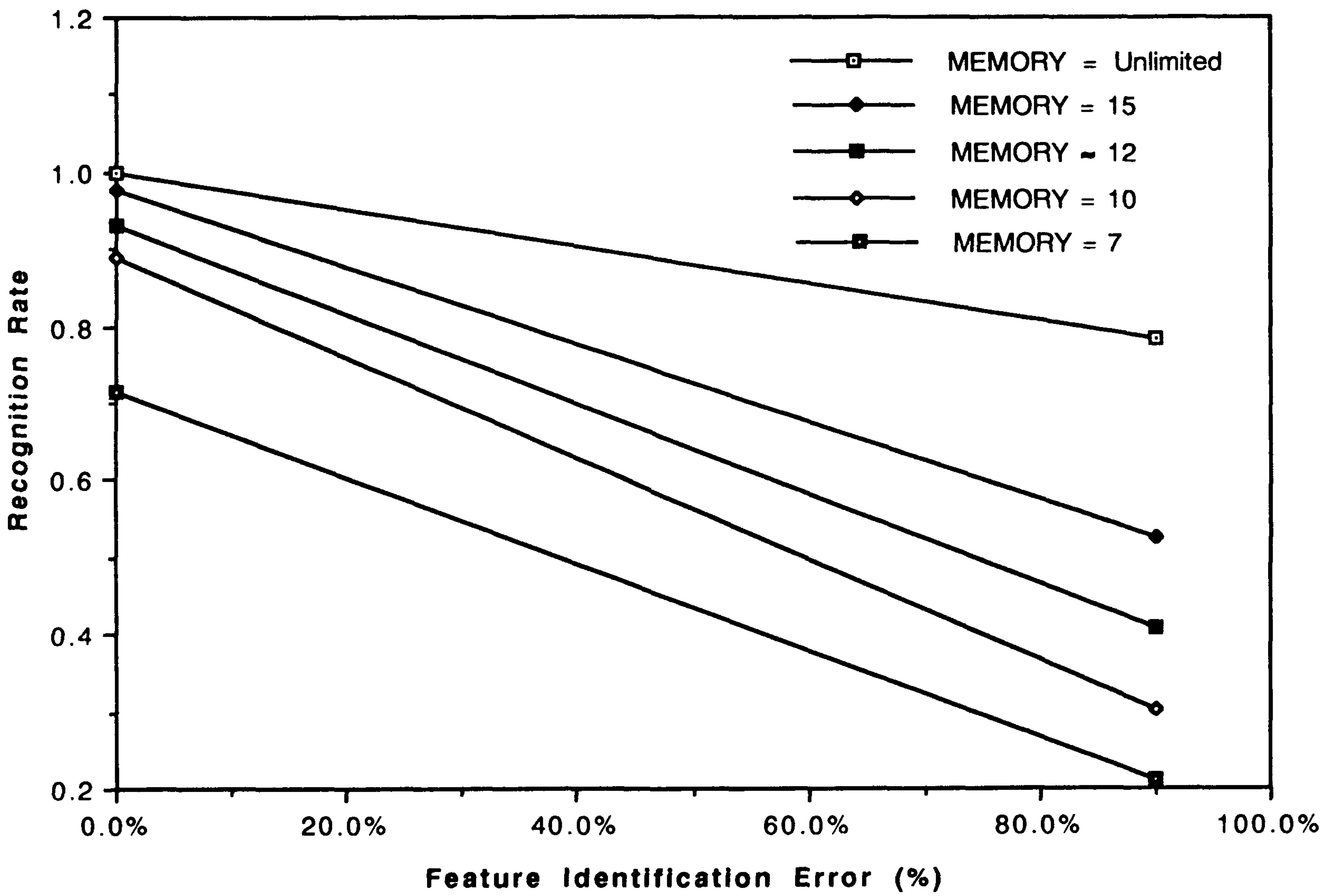


Figure 20. Results of Memory Experiment and Computer Simulation Combined

probing experiments using human subjects. Given the nature of the assumptions used to derive the computer simulation, the sacrifice of precision made in arriving at these curves is considered insignificant for such purposes.

C. DISCUSSION OF RESULTS

1. Memory Experiment

Primary memory capacity has been experimentally measured at about seven items. These experiments generally deal with recall of items with little or no rehearsal time and maximum delays on the order of 30 seconds. This experiment found a memory capacity to recall 12 items after a time span of one to four minutes but involving obvious rehearsal of feature sequence. The longer time frame implies participation of the secondary memory in the process, although the sequences were clearly not committed to any long-term memory such that they could be recalled long after they were sketched. A memory capacity of 12 features seems both reasonable and consistent with the more rigorous experimental work of professional psychologists.

The memory experiment appears to have succeeded in eliminating the influence of spatial relations and feature identification error. The time spent investigating a given sequence was comparable to the time involved in investigating an alphabet character of similar number of features. Coupled with the consistent time per feature, it appears that the recognition process did not vary with the length of the sequence. Had subjects spent far more time rehearsing the longer sequences, this could not be said with confidence.

The data points which reflect errors for sequences shorter than the cut-off value can be attributed in part to the fact that most of these sequences were among the earliest presented. Performance may have been degraded by unfamiliarity with the procedures and the best technique to use. Furthermore, given the small statistical samples, one or two

errors significantly affects the results. It is certainly expected that mistakes will occur for sequences of any length , but it is clear that beyond the cut-off values, ability to correctly recall dropped dramatically and unquestionably.

The importance of the location of errors within the sequences lies in the support given the notion that, in haptic probing, two-dimensional objects are treated as individual features connected in series. If the Gestalt approach to perception of an object as a whole was applied here one would expect errors to be distributed evenly throughout the sequences. However, if the perception is of features connected in series, limited by memory, one would expect the sequence to be correctly recalled to the limit of memory. This would result in errors in the beginning or end of the sequence, just as this experiment shows, depending on which end of the sequence had received the most recent or most concentrated attention.

2. Computer Simulation

The results described above show that for the computer simulation of the Finite Memory Model, a clear linear relationship exists between recognition rate and feature identification error for a given length of memory. Practically speaking, this means that this model predicts an inverse linear relationship between the ability of a person to recognize an object using haptic senses and the number of object features which he misidentifies. In addition, recognition rate falls with decreasing capacity to remember sequences of features. While it certainly seems reasonable that recognition rate should fall with increasing feature identification error and decreasing memory, the linear relationship was not a foregone conclusion. There was anecdotal evidence from Driels' haptic recognition experiments that recognition rate dropped dramatically and suddenly when subjects had difficulty identifying features due to high friction. Thus a recognition rate curve with a sudden drop at some critical value of feature identification error might have been expected. If haptic recognition

under these circumstances were indeed subject to such a step drop in performance, the value at which the step occurs could be treated as a critical limit for feature identification error in designs for telepresence systems. Exceeding the critical value by even a small amount would lead to serious loss of haptic recognition performance. With a linear relationship, the implication is that no such critical value exists and marginal increases in feature identification result in only marginal degradation of recognition performance.

A rigorous evaluation of the effectiveness of this model will have to await completion of comparable experiments in haptic object recognition using human subjects. However, it is possible to make some tentative comparisons with experimental data from previous work. Experiments in haptic recognition of raised two-dimensional letters by Acosta [Acosta 91] and Driels and Spain [Driels 90] provide what amounts to two data points for comparison with the recognition rate curves in Figure 20. Acosta's work used the same low friction apparatus used in this study for the memory length experiment. His work, and preliminary experiments conducted as background for this research, showed very low feature identification error and near perfect recognition rate. This correlates well with the curves in Figure 19 for low feature identification error. Driels and Spain used a similar experimental arrangement involving much greater friction between the probe and taskboard, which should correlate to a high degree of feature identification error. They reported a recognition rate of 0.4 which cannot be correlated with the computer simulation results without an objective measure of the amount of feature identification error involved. Their result does not, however, contradict the results of the computer simulation of the Finite Memory Model.

It should be evident that these results apply specifically to the particular object set chosen. As objects are made more similar, fewer errors in feature identification will be needed to confuse the identity of a particular object. If the objects to be recognized were

more similar than the set of alphabet letters used here, recognition rate should fall more rapidly with increasing feature identification error and decreasing memory length. Conversely, a set of objects that are very unique and share few of the same features should be much less sensitive to either feature identification error or memory length.

The relatively high recognition rates for the alphabet characters, even for very high feature identification error, results from two factors. First, the uniqueness of the sequence that makes up each character is still preserved to a degree, despite many misidentified features, because the substituted feature is related to the correct feature by being one of only two or three possible choices which might be reasonably confused. Clearly, a computer will be better than a human at keeping track of all these possibilities. Secondly, and most importantly, the recognition rate is based entirely on best matches with the least number of feature errors. The least number of errors could be one or it could be fifteen. While the computer can check every option and report the number of matches that can be made by changing any fifteen features, a human would have difficulty finding any possible matches with so many errors. Both of these factors result in a prediction of haptic recognition performance likely to considerably exceed actual human performance. Perhaps memory not only limits the number of sequential features which can be recalled, but also limits the process of matching error-filled sequences with the library representations. This would imply a modification of the Finite Memory Model to accomodate a memory input to the cognitive matching process, but would not nullify the basis of the model or its dependence on the two variables: memory and feature identification error. Again these questions await the completion of comparable human subject experiments.

VI. CONCLUSIONS AND RECOMMENDATIONS.

- The capacity of a human telemanipulator operator to remember a sequence of features, which define a raised two-dimensional object, identified through the proprioceptive component of the haptic sense, varies between individuals. A reasonable value for modelling purposes is about twelve (12) features. This value accounts for the entire memory process as it functions in actual haptic investigation of an object.

- The Finite Memory Model defines a haptic perception process in which matching of haptic sensory input to internal representations of known objects is limited by two factors, the capacity to correctly identify features and the capacity to remember the features and their sequence.

- The computer simulation is a faithful representation of the theoretical Finite Memory Model, incorporating a matching process limited by the same two factors of memory and feature identification error.

- In order to validate the Finite Memory Model and its computer simulation as reasonable models of actual human haptic perception, experimental data documenting human haptic recognition performance for the same object set (alphabet characters) used in the computer simulation is required. Ideally, a plot of recognition rate versus feature identification error is desired for comparison with the results of this study (Figure 20). A series of experiments, in which human subjects using a telemanipulator attempt to recognize objects (letters of the alphabet), would produce the necessary data, provided that feature identification error could be varied in a controlled manner. A preliminary experiment to measure feature identification error for various conditions of mechanical properties of the telemanipulator, such as friction, inertia and stiffness, would yield several physically different mechanical arrangements (i.e., different probe or task board material), each with a known feature identification error. By repeating the object recognition experiment for each mechanical arrangement, a set of data points would be obtained for comparison to the computer simulation results. An additional benefit of the preliminary experiment is

the knowledge gained concerning the relative importance of various mechanical properties in designing a telemanipulator to minimize feature identification error, certainly a requirement in future telepresence applications.

APPENDIX A

COMPUTER

PROGRAMS

```
'                    PROGRAM INBUILD.BAS
'
'REMARKS:   PROGRAM INBUILD.BAS CREATES INPUT FILES
'           IN(?).DAT FOR A SINGLE VALUE OF % FEATURE
'           IDENTIFICATION ERROR (PCTERR). THE INPUT FILE
'           IS COMPOSED OF EACH OBJECT STRING IN THE LIBRARY
'           (IN THIS CASE SUB ALPHABET) CORRUPTED BY THE
'           DESIRED PCTERR.  EACH INPUT FILE WILL CONTAIN A
'           LINE OF DATA FOR EACH OBJECT STRING IN THE
'           LIBRARY INCLUDING:  RUN NUMBER (RUN$), CORRUPTED
'           OBJECT STRING (DATA$),THE NAME OF THE OBJECT
'           (DATACHAR$), AND THE NUMBER OF FEATURES WHICH
'           HAVE BEEN CHANGED TO ACHIEVE THE REQUIRED PCTERR
'           (NUMBER).  THESE INPUT FILES CAN BE USED ALONE
'           WITH PROGRAM MATCH.BAS OR COMBINED IN SERIES
'           WITH SERINBUILD.BAS FOR USE WITH SERMATCH.BAS.
'
'DIRECTIONS:  REPLACE THE ? IN LINE 1000 WITH SUITABLE
'             CHARACTERS TO DESIGNATE THE INPUT FILE YOU
'             ARE CREATING.  REPLACE THE ? IN LINE 1100
'             WITH THE INTEGER VALUE OF THE PERCENTAGE
'             FEATURE IDENTIFICATION ERROR (IE 40).
'             EXECUTE THE PROGRAM.

DECLARE SUB FEATURE1 (C$, BB$(), W)
DECLARE SUB ALPHABET (M, N, ALPHA$)
DECLARE SUB SEQUENCE (M, N, A$())
CLS
OPTION BASE 0
DIM A$(27, 3)
DIM BB$(60)
DIM VV(60)
1000 OPEN "IN(?).DAT" FOR OUTPUT AS #1
1100 PCTERR = ?
     PRINT "CREATING IN("; PCTERR; ").DAT FILE, WITH ";
PCTERR; "% ERROR"
     FOR M = 1 TO 26
     IF M = 15 GOTO 3100
     FOR N = 1 TO 2
     CALL ALPHABET(M, N, ALPHA$)
     CALL SEQUENCE(M, N, A$())
```

```
      IF A$(M, N) = "" THEN GOTO 3000
      LENGTHA = LEN(A$(M, N))
      LENGTH = (LENGTHA + 1) / 2
      NUMBER = INT((LENGTH * PCTERR * .01) + .5)
      RANDOMIZE TIMER
      U = INT(RND * (LENGTH - 1)) + 1
      DATA$ = MID$(A$(M, N), U, LENGTH)
      IF NUMBER = 0 THEN GOTO 2100
      FOR K = 1 TO NUMBER
      VV(K) = 0
      NEXT K
      FOR I = 1 TO NUMBER
        RANDOMIZE TIMER
2000    V = INT(RND * (LENGTH - 1)) + 1
        FOR J = 1 TO I
          IF NUMBER = LENGTH THEN GOTO 2050
          IF V = VV(J) THEN GOTO 2000
        NEXT J
2050    C$ = MID$(DATA$, V, 1)
        CALL FEATURE1(C$, BB$(), W)
        RANDOMIZE TIMER
        B$ = BB$(INT(RND * (W - 1)) + 1)
        DATA$ = LEFT$(DATA$, V - 1) + B$ + RIGHT$(DATA$,
LENGTH - V)
        VV(I) = V
      NEXT I
2100  IF N = 1 THEN NN$ = "a"
      IF N = 2 THEN NN$ = "b"
      RUN$ = STR$(M) + NN$
      DATACHAR$ = ALPHA$
      Q$ = CHR$(34)
      PRINT #1, Q$; RUN$; Q$; Q$; DATA$; Q$; Q$; DATACHAR$;
Q$; NUMBER
3000 NEXT N
3100 NEXT M
CLOSE #1
END

'                    SUBPROGRAM ALPHABET
'
'REMARKS:  THIS SUBPROGRAM DEFINES THE NAMES OF THE OBJECT
'          STRINGS IN THE LIBRARY (SUB SEQUENCE).
'
5000 SUB ALPHABET (M, N, ALPHA$) STATIC
     IF M = 0 THEN ALPHA$ = ""
     IF N = 0 THEN ALPHA$ = ""
5010 IF M = 1 THEN ALPHA$ = "A"
5020 IF M = 2 THEN ALPHA$ = "B"
5030 IF M = 3 THEN ALPHA$ = "C"
5040 IF M = 4 THEN ALPHA$ = "D"
5050 IF M = 5 THEN ALPHA$ = "E"
5060 IF M = 6 AND N = 1 THEN ALPHA$ = "F(CW)"
5070 IF M = 6 AND N = 2 THEN ALPHA$ = "F(CCW)"
```

```
5080 IF M = 7 AND N = 1 THEN ALPHA$ = "G(CW)"
5090 IF M = 7 AND N = 2 THEN ALPHA$ = "G(CCW)"
5100 IF M = 8 THEN ALPHA$ = "H"
5110 IF M = 9 THEN ALPHA$ = "I"
5120 IF M = 10 AND N = 1 THEN ALPHA$ = "J(CW)"
5130 IF M = 10 AND N = 2 THEN ALPHA$ = "J(CCW)"
5140 IF M = 11 AND N = 1 THEN ALPHA$ = "K(CW)"
5150 IF M = 11 AND N = 2 THEN ALPHA$ = "K(CCW)"
5160 IF M = 12 AND N = 1 THEN ALPHA$ = "L(CW)"
5170 IF M = 12 AND N = 2 THEN ALPHA$ = "L(CCW)"
5180 IF M = 13 THEN ALPHA$ = "M"
5190 IF M = 14 THEN ALPHA$ = "N"
5200 IF M = 15 THEN ALPHA$ = "O"
5210 IF M = 16 AND N = 1 THEN ALPHA$ = "P(CW)"
5220 IF M = 16 AND N = 2 THEN ALPHA$ = "P(CCW)"
5230 IF M = 17 AND N = 1 THEN ALPHA$ = "Q(CW)"
5240 IF M = 17 AND N = 2 THEN ALPHA$ = "Q(CCW)"
5250 IF M = 18 AND N = 1 THEN ALPHA$ = "R(CW)"
5260 IF M = 18 AND N = 2 THEN ALPHA$ = "R(CCW)"
5270 IF M = 19 THEN ALPHA$ = "S"
5280 IF M = 20 THEN ALPHA$ = "T"
5290 IF M = 21 THEN ALPHA$ = "U"
5300 IF M = 22 THEN ALPHA$ = "V"
5310 IF M = 23 THEN ALPHA$ = "W"
5320 IF M = 24 THEN ALPHA$ = "X"
5330 IF M = 25 THEN ALPHA$ = "Y"
5340 IF M = 26 THEN ALPHA$ = "Z"
5350 END SUB

'                   SUBPROGRAM FEATURE1
'
'REMARKS:   THIS SUBPROGRAM DEFINES THE LIKELY FEATURE MIS-
'           IDENTIFICATIONS THAT MIGHT BE MADE BY A PERSON
'           SCANNING AN OBJECT WITH THE HAPTIC SENSES.  (IE
'           A RIGHT ANGLE "1" MIGHT BE MISTAKEN FOR AN ACUTE
'           ANGLE "3" OR OBTUSE ANGLE "5" BUT COULD NOT BE
'           MISTAKEN FOR A LONG STRAIGHT SIDE "L").  THIS
'           INFORMATION IS USED TO PROVIDE THE CORRUPTED
'           INPUT DATA FOR GIVEN VALUES OF % FEATURE
'           IDENTIFICATION ERROR.  CORRUPTED FEATURES CAN
'           ONLY BE CHANGED TO A FEATURE FOR WHICH IT MIGHT
'           BE MISTAKEN.
'
SUB FEATURE1 (C$, BB$(), W) STATIC
IF C$ = "1" THEN
   W = 2
   BB$(1) = "3"
   BB$(2) = "5"
ELSEIF C$ = "2" THEN
   W = 2
   BB$(1) = "4"
   BB$(2) = "6"
ELSEIF C$ = "3" THEN
```

```
    W = 2
    BB$(1) = "1"
    BB$(2) = "5"
ELSEIF C$ = "4" THEN
    W = 2
    BB$(1) = "2"
    BB$(2) = "6"
ELSEIF C$ = "5" THEN
    W = 2
    BB$(1) = "1"
    BB$(2) = "8"
ELSEIF C$ = "6" THEN
    W = 2
    BB$(1) = "2"
    BB$(2) = "0"
ELSEIF C$ = "7" THEN
    W = 3
    BB$(1) = "8"
    BB$(2) = "L"
    BB$(3) = "5"
ELSEIF C$ = "8" THEN
    W = 3
    BB$(1) = "7"
    BB$(2) = "M"
    BB$(3) = "5"
ELSEIF C$ = "9" THEN
    W = 3
    BB$(1) = "0"
    BB$(2) = "L"
    BB$(3) = "6"
ELSEIF C$ = "0" THEN
    W = 3
    BB$(1) = "9"
    BB$(2) = "2"
    BB$(3) = "6"
ELSEIF C$ = "L" THEN
    W = 3
    BB$(1) = "M"
    BB$(2) = "5"
    BB$(3) = "6"
ELSEIF C$ = "M" THEN
    W = 4
    BB$(1) = "L"
    BB$(2) = "5"
    BB$(3) = "6"
    BB$(4) = "S"
ELSEIF C$ = "S" THEN
    W = 2
    BB$(1) = "M"
    BB$(2) = "0"
END IF
END SUB
```

```
'                    SUBPROGRAM SEQUENCE
'
'REMARKS:  THIS SUBPROGRAM SERVES AS A LIBRARY OF OBJECTS
'          DESCRIBED BY STRINGS OF FEATURES EACH OF WHICH
'          IS DEFINED BY A CHARACTER. THE OBJECT SET IN
'          THIS CASE IS THE ENGLISH ALPHABET.  SYMETRICAL
'          LETTERS ARE DESCRIBED ONCE, WHILE UNSYMMETRICAL
'          LETTERS ARE DESCRIBED FIRST IN THE CLOCKWISE
'          DIRECTION AND THEN COUNTERCLOCKWISE.  THE
'          FEATURES ARE DEFINED AS FOLLOWS:
'
'                        1 = OUTSIDE RIGHT ANGLE
'                        2 = INSIDE RIGHT ANGLE
'                        3 = OUTSIDE ACUTE ANGLE
'                        4 = INSIDE ACUTE ANGLE
'                        5 = OUTSIDE OBTUSE ANGLE
'                        6 = INSIDE OBTUSE ANGLE
'                        7 = LARGE OUTSIDE CURVE
'                        8 = SMALL INSIDE CURVE
'                        9 = LARGE INSIDE CURVE
'                        0 = SMALL INSIDE CURVE
'                        L = LONG STRAIGHT SIDE
'                        M = MEDIUM STRAIGHT SIDE
'                        S = SHORT STRAIGHT SIDE
'
'          A DASH SEPARATES THE INSIDE FEATURES OF A
'          LETTER FROM THOSE ON THE OUTSIDE PERIMETER.
'          THE INSIDE PORTION OF THE STRING HAS BEEN
'          COMMENTED OUT BECAUSE EACH LETTER CAN BE
'          UNIQUELY IDENTIFIED THE OUTSIDE ALONE AND
'          BECAUSE OF DIFFICULTIES ADDRESSING THE
'          TRANSITION FROM OUTSIDE TO INSIDE.
'
SUB SEQUENCE (M, N, A$()) STATIC
     FOR I = 0 TO 26
       A$(I, 0) = ""
     NEXT I
     A$(0, 1) = ""
     A$(0, 2) = ""
6020 A$(1, 1) = "3L5S5L3S5M6M6M5S3L5S5L3S5M6M6M5"
'-4M4M4S4M4M4"
6030 A$(1, 2) = ""
6040 A$(2, 1) = "1L18481L184"   '-2M202M20"
6050 A$(2, 2) = ""
6060 A$(3, 1) = "71S191S171S191S"
6070 A$(3, 2) = ""
6080 A$(4, 1) = "1L171L1"    '-2M292M2"
6090 A$(4, 2) = ""
6100 A$(5, 1) = "1L1M1S1M2S2S1S1S2S2M1S1M1L1M1S1M2S2S1S1S
2S2M1S1"
6110 A$(5, 2) = ""
6120 A$(6, 1) = "1L1M1S1M2S2S1S1S2M1S1L1M1S1M2S2S1S1S2M1"
6130 A$(6, 2) = "S1M2S1S1S2S2M1S1M1L1S1M2S1S1S2S2M1S1M1L"
```

```
6140 A$(7, 1) = "71S192S1S1S1M1S1S471S192S1S1S1M1S1S"
6150 A$(7, 2) = "4S1S1M1S1S1S291S174S1S1M1S1S1S291S1"
6160 A$(8, 1) = "1L1S1M2S2M1S1L1S1M2S2M1S1L1S1M2S2M1S1L1S
1M2S2M1"
6170 A$(8, 2) = ""
6180 A$(9, 1) = "1L1S1L1S1L1S1L1"
6190 A$(9, 2) = ""
6200 A$(10, 1) = "81S10M1S1M81S10M1S1"
6210 A$(10, 2) = "M1S1M01S18M1S1M01S1"
6220 A$(11, 1) = "1L1S1M4M5S3M6M3S5M4S6S1S1L1S1M4M5S3M6M3
S5M4S6S1"
6230 A$(11, 2) = "S1S6S4M5S3M6M3S5M4M1S1L1S1S6S4M5S3M6M3S
5M4M1S1L"
6240 A$(12, 1) = "1L1S1L2M1S1M1L1S1L2M1S1"
6250 A$(12, 2) = "M1S1M2L1S1L1M1S1M2L1S1L"
6260 A$(13, 1) = "1L1S5M4M5S1L1S1M4M5S5M4M1S1L1S5M4M5S1L1
S1M4M5S5M4M1"
6270 A$(13, 2) = ""
6280 A$(14, 1) = "1L1S5M4M1S1L1S5M4M1S1L1S5M4M1S1L1S5M4M1"
6290 A$(14, 2) = ""
6300 A$(15, 1) = "7"      '-9"
6310 A$(15, 2) = ""
6320 A$(16, 1) = "1L182M1S1L182M1"     '-2M202M20"
6330 A$(16, 2) = "S1M281L1S1M281L"     '-02M202M2"
6340 A$(17, 1) = "76S5S3S476S5S3S"     '-94S3S5S694S3S5S"
6350 A$(17, 2) = "4S3S5S674S3S5S6"     '-6S5S3S496S5S3S4"
6360 A$(18, 1) = "1L186M3S5M6S2M1S1L186M3S5M6S2M1"
'-2M202M20"
6370 A$(18, 2) = "S1M2S6M5S3M681L1S1M2S6M5S3M681L"
'-02M202M2"
6380 A$(19, 1) = "1S1081S1081S1081S10"
6390 A$(19, 2) = ""
6400 A$(20, 1) = "1L2S1S1M1S1S2L1S1L2S1S1M1S1S2L1"
6410 A$(20, 2) = ""
6420 A$(21, 1) = "8M1S1M0M1S1M8M1S1M0M1S1"
6430 A$(21, 2) = ""
6440 A$(22, 1) = "5L3S5M4M5S3L5S5L3S5M4M5S3L5"
6450 A$(22, 2) = ""
6460 A$(23, 1) = "5L3S5M4M5S5M4M5S5L5S5M4M5S5L3S5M4M5S5M4
M5S5L5S5M4M5"
6470 A$(23, 2) = ""
6480 A$(24, 1) = "3M6M3S5M4M5S3M6M3S5M4M5S3M6M3S5M4M5S3M6
M3S5M4M5"
6490 A$(24, 2) = ""
6500 A$(25, 1) = "1M6M3S5M4M5S3M6M1S1M6M3S5M4M5S3M6M1"
6510 A$(25, 2) = ""
6520 A$(26, 1) = "1S5M4M1S1M1S5M4M1S1M1S5M4M1S1M1S5M4M1S1"
6530 A$(26, 2) = ""
END SUB
```

```
'                    PROGRAM MATCH.BAS
'
'REMARKS:   PROGRAM MATCH.BAS TAKES INPUT FILE IN(?).DAT
'           (FOR A SINGLE VALUE OF FEATURE IDENTIFICATION
'           ERROR) AND A SINGLE VALUE OF MEMORY (ENTERED
'           INTERACTIVELY) AND GIVES THE OBJECT RECOGNITION
'           RATE IN OUTPUT FILE OUT(?).DAT
'DIRECTIONS:   REPLACE THE ? IN LINES 1540 AND 1541 WITH
'              THE DESIGNATIONS OF THE DESIRED INPUT AND
'              OUTPUT FILES.  EXECUTE THE PROGRAM.  YOU
'              WILL BE ASKED TO ENTER THE LENGTH OF MEMORY
'              IN NUMBER FEATURES.
'
DECLARE SUB FEATURE (C$, B$, INDEX)
DECLARE SUB ALPHABET (M, N, ALPHA$)
DECLARE SUB SEQUENCE (M, N, A$())

990 CLS
1000 OPTION BASE 0
1010 DIM A$(26, 2)
1011 DIM ALPHA1$(2)
     DIM COUNT(30, 30)
     DIM POSSMATCH$(30, 30)

1012 CALL SEQUENCE(M, N, A$())
     SUM = 0
     RUNCOUNT = 0
1540 OPEN "IN(?).DAT" FOR INPUT AS #1
1541 OPEN "OUT(?).DAT" FOR OUTPUT AS #2
     WRITE "ENTER THE LENGTH OF MEMORY "
     INPUT MEMORY
     WHILE NOT EOF(1)
     INPUT #1, RUN$, DATA$, DATACHAR$, X
     PRINT "RUN NUMBER "; RUN$
     PRINT #2, "RUN NUMBER "; RUN$
     PRINT "DATA$ = "; DATA$; ", CHARACTER IS ";
DATACHAR$; ", MEMORY = "; MEMORY; ", X = "; X
     PRINT #2, "DATA$ = "; DATA$; ", CHARACTER IS ";
DATACHAR$; ", MEMORY = "; MEMORY; ", X = "; X
1620 LENGTH = LEN(DATA$)
1628 IF LENGTH <= MEMORY THEN LIMIT = LENGTH
1629 IF LENGTH > MEMORY THEN LIMIT = MEMORY

     FOR M = 1 TO 26
     FOR L = 0 TO X
     POSSMATCH$(L, M) = ""
     COUNT(L, M) = 0
     NEXT L
     FOR N = 1 TO 2
     IF A$(M, N) = "" THEN GOTO 1720
' CHOOSE ONE LETTER(OBJECT) FROM A$ SUBPROGRAM
     CALL ALPHABET(M, N, ALPHA$)
IF N = 1 THEN PRINT ALPHA$;
```

```
     FOR K = 1 TO LEN(DATA$) - LIMIT + 1
     MDATA$ = MID$(DATA$, K, LIMIT)
' SUCCESSIVELY LOOK AT LIMIT SIZED SEGMENTS OF THE INPUT
'DATA$
     FOR I = 1 TO LEN(A$(M, N)) - LIMIT
     MISMATCH = 0
     MATCH = 0
     AA$ = MID$(A$(M, N), I, LIMIT)
'SUCCESSIVELY LOOK AT LIMIT SIZED SEGMENTS OF LIBRARY
'STRING A$
     INDEX = 0
     FOR J = 1 TO LIMIT
     C$ = MID$(AA$, J, 1)
     B$ = MID$(MDATA$, J, 1)
     IF B$ = C$ THEN
'COMPARE THE LIMIT SIZED SEGMENTS OF DATA$ AND A$ ONE
'FEATURE AT A TIME
        MATCH = MATCH + 1
'COUNT THE NUMBER OF FEATURES THAT MATCH
     ELSE
        MISMATCH = MISMATCH + 1
'COUNT THE NUMBER OF FEATURES THAT DON'T MATCH
        CALL FEATURE(C$, B$, INDEX)
'CHECK IF MISMATCHED FEATURE IS COMPATIBLE
     END IF
     IF MISMATCH > X GOTO 1700
'IF FEATURES THAT DONT MATCH EXCEED MAX ERRORS IN DATA$
'SKIP TO NEXT
'SEGMENT
     NEXT J
     IF INDEX = LIMIT - MATCH THEN
'IF ALL MISMATCHED FEATURES ARE COMPATIBLE
     MISID$ = "LIKELY"
'IF SOME OF THE MISMATCHED FEATURES ARE NOT COMPATIBLE
     ELSE MISID$ = "UNLIKELY"
     END IF
     IF MATCH >= LIMIT - X AND MISID$ = "LIKELY" THEN
'IF MISMATCHED FEATURES ALL COMPATIBLE AND DONT EXCEED
'MAX ERRORS IN DATA$.  NOTE:THIS ALLOWS MAX ERRORS IN
'LIMIT SIZED SEGMENT RATHER THAN ENTIRE LENGTH
       PRINT #2, ALPHA$; " "; LIMIT - MATCH; " ";
AA$; " "; MDATA$
        X = LIMIT - MATCH
'MAX ERRORS TO CONSIDER REDUCED TO SMALLEST NUMBER
'ALREADY ENCOUNTERED TO SAVE TIME
        COUNT(X, M) = COUNT(X, M) + 1
     END IF
1700 NEXT I
     NEXT K
     FOR L3 = 0 TO X
     IF COUNT(L3, M) > 0 THEN
        POSSMATCH$(L3, M) = LEFT$(ALPHA$, 1)
     END IF
```

```
'IF ANY SEGMENT MATCHED (FOR NR OF ERRORS FROM 0 TO X)
'RECORD THE LIBRARY CHARACTER IDENTIFICATION
     NEXT L3
     NEXT N
1720 NEXT M
     RIGHT = 0
     WRONG = 0
     FOR MM = 1 TO 26
     IF POSSMATCH$(X, MM) = LEFT$(DATACHAR$, 1) THEN
        RIGHT = COUNT(X, MM)
     ELSE WRONG = WRONG + COUNT(X, MM)
     END IF
IF MM = 26 THEN
PRINT #2, "RIGHT = "; RIGHT; "   WRONG = "; WRONG
END IF
     NEXT MM
     SCORE = RIGHT / (WRONG + RIGHT)
     PRINT "SCORE FOR RUN "; RUN$; " = "; SCORE
     PRINT #2, "SCORE FOR RUN "; RUN$; " = "; SCORE
1906 RUNCOUNT = RUNCOUNT + 1
     SUM = SUM + SCORE
     PRINT #2,
     PRINT
     WEND
     RECRATE = SUM / RUNCOUNT
     PRINT #2, "RECOGNITION RATE FOR "; RUNCOUNT;
" INPUT DATA STRINGS = ";RECRATE
1924 CLOSE #1
     CLOSE #2
1930 END

'                     SUBPROGAM FEATURE
'
'REMARKS:  THIS SUBPROGRAM IS USED IN MATCHING THE
'          CORRUPTED INPUT DATA STRINGS WITH THE LIBRARY
'          OF OBJECTS (IN SUB SEQUENCE).  AS EACH FEATURE
'          IS COMPARED A CHECK IS MADE TO SEE THAT IF NOT
'          THE SAME FEATURE AT LEAST IT IS AMONG THOSE
'          WHICH MIGHT BE EASILY CONFUSED OR MISIDENTIFIED.
'
SUB FEATURE (C$, B$, INDEX) STATIC
IF C$ = "1" AND (B$ = "3" OR B$ = "5") THEN
   INDEX = INDEX + 1
ELSEIF C$ = "2" AND (B$ = "4" OR B$ = "6") THEN
   INDEX = INDEX + 1
ELSEIF C$ = "3" AND (B$ = "1" OR B$ = "5") THEN
   INDEX = INDEX + 1
ELSEIF C$ = "4" AND (B$ = "2" OR B$ = "6") THEN
   INDEX = INDEX + 1
ELSEIF C$ = "5" AND (B$ = "1" OR B$ = "8") THEN
   INDEX = INDEX + 1
ELSEIF C$ = "6" AND (B$ = "2" OR B$ = "0") THEN
   INDEX = INDEX + 1
```

```
ELSEIF C$ = "7" AND (B$ = "8" OR B$ = "L" OR B$ = "5")
THEN
   INDEX = INDEX + 1
ELSEIF C$ = "8" AND (B$ = "7" OR B$ = "M" OR B$ = "5")
THEN
   INDEX = INDEX + 1
ELSEIF C$ = "9" AND (B$ = "0" OR B$ = "L" OR B$ = "6")
THEN
   INDEX = INDEX + 1
ELSEIF C$ = "0" AND (B$ = "9" OR B$ = "2" OR B$ = "6")
THEN
   INDEX = INDEX + 1
ELSEIF C$ = "L" AND (B$ = "M" OR B$ = "5" OR B$ = "6")
THEN
   INDEX = INDEX + 1
ELSEIF C$ = "M" AND (B$ = "L" OR B$ = "5" OR B$ = "6"
OR B$ = "S") THEN
   INDEX = INDEX + 1
ELSEIF C$ = "S" AND (B$ = "M" OR B$ = "0") THEN
   INDEX = INDEX + 1
END IF
END SUB

'               PROGRAM SERINBLD.BAS
'
'REMARKS:  SERINBLD.BAS COMBINES A SERIES OF IN(?).DAT
'          FILES FOR VARIOUS PERCENT FEATURE IDENTIFICATION
'          ERROR (IE 0%, 5%, 10%, ETC)INTO A SINGLE INPUT
'          FILE IN(SER?).DAT WHICH IS READ BY THE PROGRAM
'          SERMATCH.BAS.  THE RANGE OF % FEATURE ID ERROR
'          TO USE SHOULD BE SUCH THAT A SINGLE RUN OF
'          SERMATCH.BAS WILL YIELD ENOUGH DATA POINTS TO
'          PRODUCE A COMPLETE CURVE OF OBJECT RECOGNITION
'          RATE VS % FEATURE ID ERROR FOR A SINGLE VALUE
'          OF MEMORY.
'DIRECTIONS:  REPLACE THE ? IN LINE 3 WITH A SUITABLE
'             CHARACTER AND ENSURE THAT THE FILE SO NAMED
'             IS EMPTY (IF IN DOUBT ERASE ANY FILE BY THAT
'             NAME). REPLACE THE ? IN LINE 2 WITH THE
'             CHARACTERS DESIGNATING THE FIRST OF THE
'             BASIC INPUT FILES TO BE COMBINED IN SERIES.
'             EXECUTE THE PROGRAM.  CHANGE LINE 2
'             SUCCESSIVELY TO ADD THE REMAINING INPUT FILES.

     CLS
2    OPEN "IN(?).DAT" FOR INPUT AS #1
3    OPEN "IN(SER?).DAT" FOR APPEND AS #2
     WHILE NOT EOF(1)
     INPUT #1, RUN$, DATA$, DATACHAR$, X
     Q$ = CHR$(34)
     PRINT #2, Q$; RUN$; Q$; Q$; DATA$; Q$; Q$;
DATACHAR$; Q$; X
     WEND
```

```
     CLOSE #1
     CLOSE #2
     END

'                 PROGRAM SERMATCH.BAS
'
'REMARKS:   PROGRAM SERMATCH.BAS IS A MODIFICATION OF
'           MATCH.BAS WHICH READS INPUT FILE IN(SER?).DAT
'           WHICH COVERS A WHOLE RANGE OF % FEATURE
'           IDENTIFICATION ERRORS (PCTERR) AND PRODUCES
'           AN OUTPUT FILE M?SER?.DAT.  THIS OUTPUT FILE
'           SHOULD YIELD ENOUGH DATA POINTS TO PLOT A CURVE
'           OF OBJECT RECOGNITION RATE VS PCTERR FOR A
'           GIVEN VALUE OF MEMORY (BY COMPARISON MATCH.BAS
'           WILL PROVIDE ONLY A SINGLE DATA POINT).
'
'DIRECTIONS:   REPLACE THE ? FOLLOWING "MEMORY = " WITH THE
'              VALUE DESIRED IN INTEGER NUMBER OF FEATURES.
'              INSERT THE SAME VALUE FOLLOWING THE "M" IN
'              THE OUTPUT FILE OPEN STATEMENT.  REPLACE THE
'              ? FOLLOWING "SER" IN BOTH THE INPUT AND
'              OUTPUT FILE OPEN STATEMENTS WITH THE
'              APPROPRIATE SERIES DESIGNATION.  EXECUTE THE
'              PROGRAM.  CAUTION:   RUN TIME FOR 10 VALUES
'              OF PCTERR (AND HENCE 10 DATA POINTS) IS
'              APPROX 24 HRS ON A 286 MACHINE.

DECLARE SUB FEATURE (C$, B$, INDEX)
DECLARE SUB ALPHABET (M, N, ALPHA$)
DECLARE SUB SEQUENCE (M, N, A$())

990 CLS
1000 OPTION BASE 0
1010 DIM A$(26, 2)
1011 DIM ALPHA1$(2)
DIM COUNT(30, 30)
DIM POSSMATCH$(30, 30)

     MEMORY = ?
     OPEN "IN(SER2).DAT" FOR INPUT AS #1
     OPEN "M?SER?.DAT" FOR OUTPUT AS #2
1012 CALL SEQUENCE(M, N, A$())
     WHILE NOT EOF(1)
     SUM = 0
     RUNCOUNT = 0
     FOR II = 1 TO 33
     INPUT #1, RUN$, DATA$, DATACHAR$, X
1620 LENGTH = LEN(DATA$)
1628 IF LENGTH <= MEMORY THEN LIMIT = LENGTH
1629 IF LENGTH > MEMORY THEN LIMIT = MEMORY

     FOR M = 1 TO 26
     FOR L = 0 TO X
```

```
     POSSMATCH$(L, M) = ""
     COUNT(L, M) = 0
     NEXT L
     FOR N = 1 TO 2
     IF A$(M, N) = "" THEN GOTO 1720
' CHOOSE ONE LETTER FROM A$ SUBPROGRAM
     CALL ALPHABET(M, N, ALPHA$)
IF N = 1 THEN PRINT ALPHA$;
     FOR K = 1 TO LEN(DATA$) - LIMIT + 1
     MDATA$ = MID$(DATA$, K, LIMIT)
' SUCCESSIVELY LOOK AT LIMIT SIZED SEGMENTS OF THE INPUT
'DATA$
     FOR I = 1 TO LEN(A$(M, N)) - LIMIT
     MISMATCH = 0
     MATCH = 0
     AA$ = MID$(A$(M, N), I, LIMIT)
'SUCCESSIVELY LOOK AT LIMIT SIZED SEGMENTS OF LIBRARY
'STRING A$
     INDEX = 0
     FOR J = 1 TO LIMIT
     C$ = MID$(AA$, J, 1)
     B$ = MID$(MDATA$, J, 1)
     IF B$ = C$ THEN
'COMPARE THE LIMIT SIZED SEGMENTS OF DATA$ AND A$ ONE
'FEATURE AT A TIME
        MATCH = MATCH + 1
'COUNT THE NUMBER OF FEATURES THAT MATCH
     ELSE
        MISMATCH = MISMATCH + 1
'COUNT THE NUMBER OF FEATURES THAT DON'T MATCH
        CALL FEATURE(C$, B$, INDEX)
'CHECK IF MISMATCHED FEATURE IS COMPATIBLE
     END IF
     IF MISMATCH > X GOTO 1700
'IF FEATURES THAT DONT MATCH EXCEED MAX ERRORS IN DATA$
'SKIP TO NEXT SEGMENT
     NEXT J
     IF INDEX = LIMIT - MATCH THEN
' F ALL MISMATCHED FEATURES ARE COMPATIBLE
     MISID$ = "LIKELY"
'IF SOME OF THE MISMATCHED FEATURES ARE NOT COMPATIBLE
     ELSE MISID$ = "UNLIKELY"
     END IF
     IF MATCH >= LIMIT - X AND MISID$ = "LIKELY" THEN
'IF MISMATCHED FEATURES ALL COMPATIBLE AND DONT EXCEED
'MAX ERRORS IN DATA$.   NOTE:THIS ALLOWS MAX ERRORS IN
'LIMIT SIZED SEGMENT RATHER THAN ENTIRE LENGTH

        X = LIMIT - MATCH
'MAX ERRORS TO CONSIDER REDUCED TO SMALLEST NUMBER
'ALREADY ENCOUNTERED TO SAVE TIME
        COUNT(X, M) = COUNT(X, M) + 1
     END IF
```

```
1700 NEXT I
     NEXT K
     FOR L3 = 0 TO X
     IF COUNT(L3, M) > 0 THEN
        POSSMATCH$(L3, M) = LEFT$(ALPHA$, 1)
     END IF
'IF ANY SEGMENT MATCHED (FOR NR OF ERRORS FROM 0 TO X)
'RECORD THE LIBRARY CHARACTER IDENTIFICATION
     NEXT L3
     NEXT N
1720 NEXT M
     RIGHT = 0
     WRONG = 0
     FOR MM = 1 TO 26
     IF POSSMATCH$(X, MM) = LEFT$(DATACHAR$, 1) THEN
        RIGHT = COUNT(X, MM)
     ELSE WRONG = WRONG + COUNT(X, MM)
     END IF
     NEXT MM
     SCORE = RIGHT / (WRONG + RIGHT)
     PRINT "SCORE FOR RUN "; RUN$; " = "; SCORE
     PRINT #2, "SCORE FOR RUN "; RUN$; " = "; SCORE
1906 RUNCOUNT = RUNCOUNT + 1
     SUM = SUM + SCORE
     PRINT
     NEXT II
     RECRATE = SUM / RUNCOUNT
     PRINT #2, "RECOGNITION RATE FOR "; RUNCOUNT;
" INPUT DATA STRINGS = "; RECRATE
     PRINT #2,
     WEND
1924 CLOSE #1
     CLOSE #2
1930 END
```

APPENDIX B

MEMORY

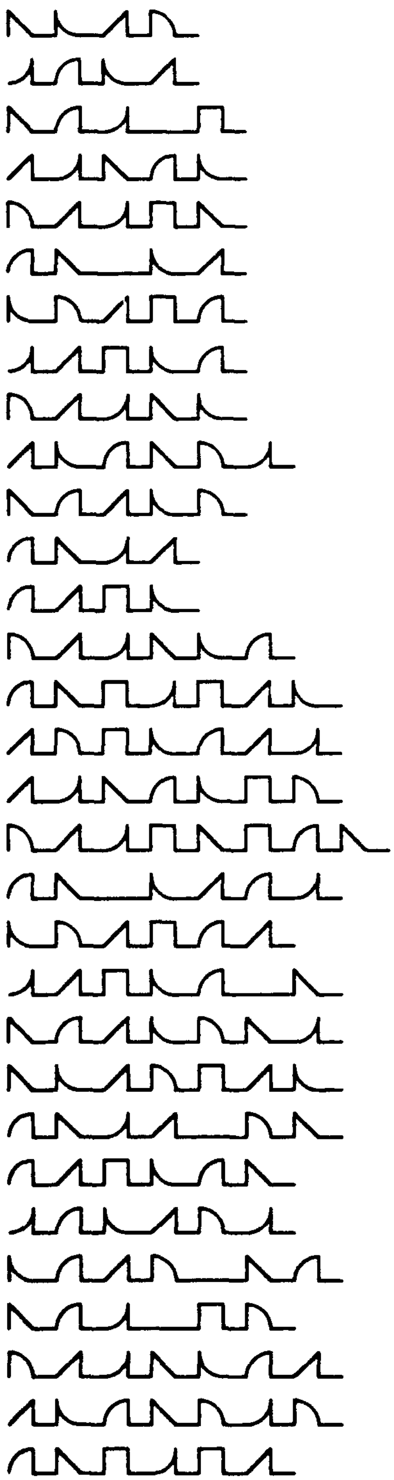
EXPERIMENT

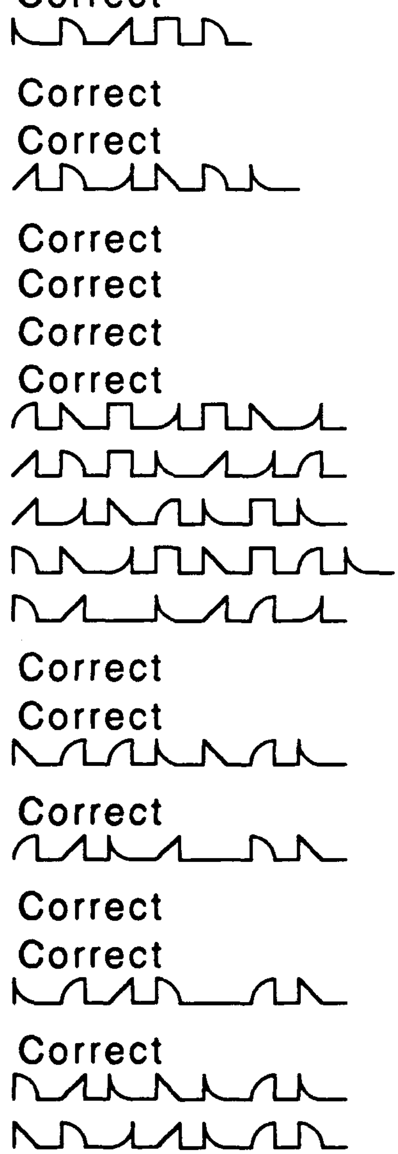
DATA

Subject Number 1

Test Sequence	Subject's Recall	Length	Time	Score
		7	1:30	0
	Correct	7	1:25	1
		7	1:15	0
	Correct	6	1:30	1
	Correct	8	1:40	1
	Correct	8	1:30	1
	Correct	7	1:29	1
	Correct	8	1:00	1
	Correct	6	1:12	1
		6	0:59	0
	Correct	7	1:35	1
	Correct	8	1:07	1
	Correct	8	1:06	1
	Correct	9	2:35	1
	Correct	10	2:37	1
	Correct	9	1:37	1
	Correct	10	1:48	1
	Correct	10	1:20	1
	Correct	10	1:15	1
		9	2:20	0
	Correct	8	N/A	1

Subject Number 1





Test Sequence	Subject's Recall	Length	Time	Score
	Correct	8	1:10	1
	Correct	8	0:50	1
	Correct	8	1:30	1
	Correct	10	1:50	1
	Correct	9	2:08	1
	Correct	9	1:25	1
		9	2:00	0
	Correct	9	1:15	1
	Correct	10	2:10	1
		12	2:30	0
	Correct	10	1:42	1
	Correct	8	1:53	1
	Correct	7	1:03	1
	Correct	12	1:50	1
		12	2:15	0
		13	2:57	0
		13	3:12	0
		14	2:55	0
		13	2:25	0
	Correct	11	1:26	1
	Correct	12	2:10	1
		14	1:35	0
	Correct	13	3:01	1
		13	2:35	0
	Correct	11	0:50	1
	Correct	12	1:40	1
		13	1:57	0
	Correct	10	1:28	1
		14	2:35	0
		14	2:45	0
	Correct	12	3:05	1

Subject Number 2

Test Sequence	Subject's Recall	Length	Time	Score
	Correct	7	1:06	1
	Correct	8	1:12	1
	Correct	10	0:59	1
	Correct	9	2:00	1
	Correct	10	1:18	1
	Correct	9	1:08	1
	Correct	12	1:35	1
	Correct	13	2:04	1
	Correct	13	2:22	1
	Correct	14	2:00	1
	Correct	13	2:25	1
	Correct	11	1:37	1
	Correct	12	1:41	1
	Correct	14	1:59	1
	Correct	13	2:06	1
	Correct	13	1:54	1
	Correct	11	1:50	1
		16	3:52	0
	Correct	13	1:57	1
		14	2:08	0
	Correct	14	4:00	1
	Correct	14	2:29	1
		14	2:12	0
		16	4:02	0
		15	2:08	0
		15	1:16	0
	Correct	15	1:54	1
		16	1:59	0
	Correct	15	1:56	1

LIST OF REFERENCES

[ROV 84] *Operational Guidelines for Remotely Operated Vehicles* , Marine Technology Society, Washington , D.C., 1984.

[Pepper 86] Pepper R., "Diver Task Description and Analysis for TOPS Project" Naval Ocean Systems Center Technical Report, Advanced Systems Division, December 1986.

[Johnsen 71] Johnsen E.G. and Corliss W.R., *Human Factors Appications in Teleoperator Design and Operation* , Wiley-Interscience, 1971.

[Driels 90] Driels M.R. and Spain H., "Haptic Recognition Through Remote Teleoperation", *Ergonomics of Hybrid Automated Systems II,* Karwowski and Rashimi, ed., Elsevier Science Publishers B.V., 1990.

[Gibson 66] Gibson J.J., *The Senses Considered as Perceptual Systems,* Houghton Mifflin Publishers, 1966.

[Klatzky 87] Klatzky R.L. and Lederman S.J., "The Intelligent Hand", *The Psychology of Learning and Motivation, Vol 21* , G. Bower, ed., New York: Academic Press 1987.

[Wolsk 67] Alpern M., Lawrence M. and Wolsk D., *Sensory Processes,* Brooks/Cole Publishing Company, Belmont, California, 1967

[Clark 86] Clark F.J., Horch K.W., "Kinesthesia", *Handbook of Perception and Human Performance* , Vol 1, Boff X.R., Kaufmann L., Thomas J.P., eds., John Wiley and Sons, New York, 1986.

[Weintraub 66] Weintraub D.J. and Walker E.L., *Perception* , Brooks/Cole Publishing Company, Belmont, California, 1966

[Stark 71] Noton D. and Stark L., "Eye Movements and Visual Perception", *Scientific American*, June 1971, Vol 224, No 6, pp 34-43.

[Stark 81] Stark L. and Ellis S.R., "Scanpaths Revisited: Cognitive Models Direct Active Looking", *Eye Movements: Cognitive and Visual Perception* , Fisher D., Monty R., and Senders J., eds., Lawrence Erlbaum Assoc., New Jersey, 1981.

[Acosta 91] Acosta J.C., *Modeling of Explorative Procedures for Remote Object Identification*", Master's Thesis, Naval Postgraduate School, Monterey, California, September 1991.

[Underwood 76] Underwood G., *Attention and Memory* , Pergamon Press, Elmsford, New York, 1976.